

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

**Návrh a implementaci modulů pro  
parametrické poptávky, vyhledávání,  
platby a tiketovací systém portálu  
Gloffer**

**Design and implementation of Gloffer  
Parametric Demand, Search, Payment  
and Ticketing Modules**

## Zadání diplomové práce

Student:

**Bc. Jan Mičanin**

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T025 Informatika a výpočetní technika

Téma:

Návrh a implementaci modulů pro parametrické poptávky, vyhledávání,  
platby a tiketovací systém portálu Gloffer  
Design and implementation of Gloffer Parametric Demand, Search,  
Payment and Ticketing Modules

Jazyk vypracování:

čeština

Zásady pro vypracování:

Cílem práce je provést analýzu, návrh a implementaci modulu pro parametrické poptávky a vyhledávání v rámci projektu Gloffer. Dále bude provedena analýza a implementace vlastního tiketovacího systému, zpracování online plateb včetně oprávnění a omezení na základě zakoupených služeb portálu Gloffer.

1. Student nastuduje a realizuje systém tvorby dynamických filtrů, parametrických poptávek a vyhledávání dle definovaných produktových kategorií za pomoci vhodné technologie (Elastic Search). Následně analyzuje, navrhne a implementuje možnosti znovu použitelnosti filtrů a výsledků vyhledávání.
2. Student provede analýzu správy uživatelských požadavků portálu Gloffer, nastuduje existující řešení a provede vlastní implementaci do portálu Gloffer, zajistí správu tiketů a komunikace s registrovanými i neregistrovanými uživateli.
3. Student popíše možná řešení v oblasti online plateb. Student implementuje napojení na platební bránu (GoPay) a internetové bankovníctví (FIO) včetně přidělení a kontroly oprávnění uživatelů a limitů na základě zakoupených služeb.
4. V závěru student provede srovnání výsledné implementace s existujícími řešeními a zhodnotí, zda je efektivní budovat vlastní moduly do informační platformy nebo využít již existující řešení a komponenty.

Seznam doporučené odborné literatury:

- [1] GORMLEY, Clinton a Zachary TONG. Elasticsearch: the definitive guide. ISBN 1449358543.
- [2] SHKLAR, Leon. a Rich. ROSEN. Web application architecture: principles, protocols and practices. 2nd ed. Hoboken, NJ: Wiley, c2009. ISBN 047051860x.
- [3] SHIVAKUMAR, Shailesh Kumar. Architecting high performing, scalable and available enterprise web applications. ISBN 9780128022580.
- [4] WEERAWARANA, Sanjiva. Web services platform architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and more. Upper Saddle River, NJ: Prentice Hall PTR, c2005. ISBN 0131488740.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Radoslav Fasuga, Ph.D.**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018



---

doc. Ing. Jan Platoš, Ph.D.  
vedoucí katedry



---

prof. Ing. Pavel Brandštetter, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární  
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 23. dubna 2018

.....  
*Melvin Sam*

Rád bych poděkoval svému vedoucímu, Ing. Radoslavu Fasugovi, Ph.D., za odbornou pomoc a konzultaci této diplomové práce, bez něhož by práce nevznikla.

## **Abstrakt**

Cílem této diplomové práce je popis problematiky parametrického vyhledávání, poptávání katalogových produktů, správy uživatelských požadavků a možností online plateb v rámci portálu Gloffer. Práce obsahuje popis možností cachování a srovnání úložišť Redis a Memcached. Součástí řešení je také návrh implementace parametrického vyhledávání, ticketovacího systému a komponenty pro objednávání a kontrolu zakoupených služeb. K implementaci je použit PHP framework Nette, Javascript, Elasticsearch, MySQL a Redis. Pro získávání informací o bankovních transakcích a provádění platebních operací bude sloužit Fio API a GoPay API. V závěru práce jsou uvedeny možnosti budoucího rozvoje řešení, srovnání s existujícími systémy a zhodnocení, zdali je lepší použít vlastní nebo již existující řešení. Implementované moduly jsou integrovány do poptávkového systému Gloffer a nasazeny do ostrého provozu.

**Klíčová slova:** Parametrické vyhledávání, poptávkový systém, ticketovací systém, online platby, cache, PHP, Nette, Elasticsearch, Redis, Fio API, GoPay API

## **Abstract**

The goal of this diploma thesis is description the problems of parametric search, demand for catalog products, managing user tickets and possibilities of online payments within the Gloffer portal. Thesis contains description possibilities of caching and comparison Redis and Memcached repositories. Part of the solution is also a proposal to implement a parametric search, ticket system and components for ordering and checking purchased services. PHP framework Nette, Javascript, Elasticsearch, MySQL and Redis are used to implement. The Fio API and the GoPay API will be used to retrieve information about bank transactions and make payment transactions. At the end of the thesis are presented possibilities of future solution development, comparison with existing systems and evaluation whether it is better to use own or existing solution. The implemented modules are integrated into the Gloffer demand system and put into operation.

**Key Words:** Parametric search, Demand system, Ticket system, Online payments, Cache, PHP, Nette, Elasticsearch, Redis, Fio API, GoPay API

# Obsah

<b>Seznam použitých zkratk a symbolů</b>	<b>9</b>
<b>Seznam obrázků</b>	<b>11</b>
<b>Seznam tabulek</b>	<b>12</b>
<b>Seznam výpisů zdrojového kódu</b>	<b>13</b>
<b>1 Úvod</b>	<b>14</b>
<b>2 Poptávkový systém Gloffer</b>	<b>16</b>
<b>3 Použité technologie</b>	<b>19</b>
3.1 Nette . . . . .	19
3.2 Composer . . . . .	19
3.3 Git . . . . .	20
3.4 MySQL . . . . .	20
3.5 Elasticsearch . . . . .	20
3.6 Redis . . . . .	21
3.7 Fio API . . . . .	21
3.8 GoPay API . . . . .	22
<b>4 Parametrické vyhledávání</b>	<b>23</b>
4.1 Rozbor parametrického vyhledávání . . . . .	23
4.2 Návrh a realizace parametrického vyhledávání . . . . .	33
4.3 Porovnání výkonu . . . . .	52
<b>5 Poptávka a nabídka</b>	<b>54</b>
5.1 Analýza poptávek . . . . .	54
5.2 Poptávky v systému Gloffer . . . . .	56
<b>6 Správa uživatelských požadavků a komunikace se zákazníky</b>	<b>59</b>
6.1 Analýza komunikace s uživateli . . . . .	59
6.2 Tiktovací systém . . . . .	59
6.3 Návrh a implementace modulu uživatelských požadavků . . . . .	66
<b>7 Online platby a objednávání služeb</b>	<b>74</b>
7.1 Analýza aktuálního stavu . . . . .	74
7.2 Návrh a implementace . . . . .	77

<b>8</b>	<b>Testování a nasazení</b>	<b>90</b>
<b>9</b>	<b>Možnosti dalšího rozvoje řešení</b>	<b>93</b>
<b>10</b>	<b>Zavěr</b>	<b>94</b>
	<b>Literatura</b>	<b>95</b>
	<b>Přílohy</b>	<b>96</b>
<b>A</b>	<b>Příloha na CD/DVD</b>	<b>97</b>



## Seznam použitých zkratk a symbolů

ACL	– Access Control List
AJAX	– Asynchronous Javascript and XML
AOF	– Append Only Files
API	– Application Programming Interface
BSD	– Berkeley Software Distribution
CRUD	– Create, Read, Update, Delete
ČSOB	– Československá obchodní banka
DBMS	– Database Management System
DPH	– Daň z přidané hodnoty
EAN	– European Article Number
EET	– Elektronická evidence tržeb
ER	– Entity relationship
FUP	– Fair User Policy
GB	– Gigabyte
GPL	– General Public Licence
HTML	– HyperText Markup Language
IMAP	– Internet Message Access Protocol
ISBN	– Internation Standard Book Number
JSON	– Javascript Object Notation
KB	– Kilobyte
MB	– Megabyte
Mpix	– Mega Pixel
MVC	– Model View Controller
MVP	– Model View Presenter
NFC	– Near Field Communication
ORM	– Object Relational Mapping
PDF	– Portable Document Format
PHP	– PHP: Hypertext Preprocessor
RAM	– Random Access Memory
RDB	– Redis Database File
REST	– Representational state transfer
SASL	– Simple Authentication and Security Layer
SDK	– Software Development Kit
SHA	– Secure Hash Algorithm
SMTP	– Simple Mail Transfer Protocol
SQL	– Structured Query Language

SSL	– Secure Sockets Layer
TTL	– Time to live
URL	– Uniform Resource Locator
UTF	– Unicode Transformation Format
XML	– eXtensible Markup Language

## Seznam obrázků

1	Domovská stránka portálu Gloffer . . . . .	17
2	Detail katalogového produktu portálu Gloffer . . . . .	18
3	Ukázka detailního parametrického filtru vyhledávání e-shopu Alza.cz . . . . .	24
4	Ukázka detailního parametrického filtru vyhledávání e-shopu CZC.cz . . . . .	25
5	Ukázka filtru vyhledávání nákupní galerie Heureka.cz . . . . .	26
6	Ukázka filtru vyhledávání cenového srovnávače Zbozi.cz . . . . .	27
7	Trend vývoje popularity technologií Memcached a Redis . . . . .	33
8	ERD model parametrického vyhledávání a kategorie produktů . . . . .	35
9	Ukázka rychlého filtru parametrického vyhledávání . . . . .	36
10	Ukázka detailního filtru parametrického vyhledávání . . . . .	37
11	Kořenové kategorie produktů portálu Gloffer . . . . .	38
12	Proces přípravy datových podkladů parametrického vyhledávání . . . . .	40
13	Proces sestavení dotazu a vykreslení výsledku vyhledávání . . . . .	51
14	Poptávková sekce portálu Gloffer . . . . .	57
15	Formulář pro vytvoření poptávky na katalogový produkt . . . . .	57
16	Dashboard systému Freshdesk . . . . .	61
17	Detail tiketu systému Freshdesk . . . . .	61
18	Dashboard systému Zendesk . . . . .	62
19	Detail tiketu systému Zendesk . . . . .	62
20	Stavový diagram požadavku . . . . .	63
21	Základní diagram případů užití tiketovacího systému . . . . .	65
22	ER diagram tiketovacího systému . . . . .	67
23	Přehled tiketů uživatele . . . . .	68
24	Detail tiketu uživatele . . . . .	69
25	Přehled tiketů v administrátorské části tiketovacího systému . . . . .	70
26	Vytvoření nového emailu prostřednictvím tiketovacího systému . . . . .	71
27	ER diagram ceníků, objednávek a transakcí . . . . .	79
28	Správa oprávnění přístupu k firmě . . . . .	80
29	Ceník a objednávkový formulář služeb . . . . .	82
30	Stavový diagram objednávky . . . . .	83
31	Faktura ve formátu PDF . . . . .	84
32	Přehled objednávek služeb . . . . .	85
33	Detail objednávky služeb . . . . .	85
34	Diagram testování a nasazení aplikace . . . . .	90
35	Ladící nástroj Tracy . . . . .	91
36	Developerské prostředí internetového prohlížeče Google Chrome . . . . .	91
37	Developerská konzole monitorovacího systému Kibana . . . . .	92

## Seznam tabulek

2	Srovnání Memcached a Redis . . . . .	31
3	Naměřené časy vykonávání při ověřování hypotézy cachování . . . . .	52

## Seznam výpisů zdrojového kódu

1	Ukázka struktury dat podkladů vyhledávání . . . . .	38
2	Ukázka metody pro odeslání dotazu pomocí knihovny Elasticsearch . . . . .	41
3	Ukázka dotazu na počet výsledků vyhledávání . . . . .	42
4	Ukázka dotazu na výchozí hodnoty parametrů . . . . .	44
5	Ukázka výsledku vyhledávání . . . . .	45
6	Ukázka konfigurace filtru . . . . .	48
7	Šablona těla e-mailu . . . . .	72
8	Ukázka části těla metody pro stahování transakcí z Fio účtu . . . . .	86
9	Dotaz časově omezené paušalovými tarify . . . . .	89
10	Dotaz na omezené kreditovými tarify . . . . .	89

# 1 Úvod

V dnešní době je prosazován trend online nakupování. Lidé chtějí nakupovat v nejkratší době, na jednom místě a za co možná nejnižší ceny. Tyto požadavky se snaží splnit nově vznikající poptávkový systém Gloffer.

Tato diplomová práce se zabývá problematikou parametrického vyhledávání, poptávek, správou uživatelských požadavků a placenými službami tohoto systému. Aby bylo možné navrhnout a implementovat jednotlivé moduly, bude nejprve nutné analyzovat dané problémy.

Parametrické vyhledávání je jedním ze základních stavebních kamenů poptávkového systému a umožňuje vytvářet poptávky na katalogové produkty. Důvodem řešení správy uživatelských požadavků je komunikace se zákazníky a získávání zpětné vazby, unifikace firemní komunikace a sjednocení více komunikačních kanálů do jednoho bodu. Dle nastaveného obchodního modelu poptávkového systému je zapotřebí analyzovat problematiku online plateb a vytvořit modul pro objednávání služeb a kontrolu přístupu k placeným službám. Diplomová práce je rozdělena dle několika modulů, přičemž jednotlivé části mají návaznost na druhé. Spojením jednotlivých částí poté vznikne celistvé řešení.

V kapitole Použité technologie se nachází popis technologií, které budou využity v praktické části této práce. U jednotlivých technologií bude uveden jejich popis, funkcionality, a případně konvence ukládání dat. Použité technologie byly vybrány tak, aby je bylo možné využít na platformě operačního systému Linux.

Kapitola Parametrické vyhledávání se zabývá analýzou, návrhem a implementací modulu pro vyhledávání založené na parametrech katalogových produktů. Nedílnou součástí analýzy jsou existující řešení na tuzemském trhu a metody formulace dotazů vyhledávání. Důraz je kladen na cachování dat, přičemž proběhne srovnání existujících technologií. Cílem praktické části je skládání dotazů do vyhledávacího systému Elasticsearch a sestavování dynamických filtrů. Kapitulu bude uzavírat porovnáním výkonu s použitím znovupoužitelného obsahu a bez něj.

V další kapitole se nachází problematika poptávek a nabídek. V analýze jsou zahrnuty definice klíčových pojmů. Analyzovány budou také existující poptávkové systémy katalogových produktů. Podkapitola Poptávky systému Gloffer obsahuje popis implementovaných typů poptávek a možná rozšíření poptávkového systému. Součástí je také průzkum, který má za úkol odhalit reálné potřeby trhu.

Další kapitolou je Správa uživatelských požadavků a komunikace se zákazníky. Analýza aktuálního stavu obsahuje definice základních pojmů, zásady komunikace se zákazníky, analýza existujících řešení a specifikace požadavků modulu. V praktické části je modul rozdělen na administrátorskou a uživatelskou část. Součástí řešení je propojení emailové komunikace s tiketovacím systémem.

Kapitola Online platby a objednávání služeb se zabývá problematikou přijímání a zpracování online plateb, napojením na internetové bankovníctví FIO banky a realizací platby kartou

prostřednictvím služby GoPay. Analýza se týká možnostmi online plateb v České republice a obchodním modelem portálu Gloffer. Podkapitoly praktické části se zabývají oprávněním uživatelů v systému, procesu objednání a zakoupení služeb. Následuje proces omezování přístupů a služeb zákazníků na základě zaplacených objednávek.

V kapitole Testování a nasazení jsou popsány způsoby testování funkcionalit a proces integrace modulů do poptávkového systému. Poslední kapitola je věnována možným vylepšením a budoucímu rozvoji řešení. V závěru této práce proběhne shrnutí práce a zhodnocení, zdali je lepší využít vlastní řešení nebo existující dostupné komponenty na trhu.

## 2 Poptávkový systém Gloffer

Provozovatelem poptávkového systému Gloffer je společnost Serious Investment s.r.o.. Společnost byla založena 8. listopadu 2012. Sídlo společnosti se nachází v Ostravě. Podnikatelská činnost se skládá z několika oblastí. Prvotní činností je prodej a investice numismatického materiálu. V této oblasti je společnost smluvním partnerem České národní banky, Národní banky Slovenska a České mincovny. Serious Investment dále působí v oblasti analýzy elektronické komerce a vývoji informačních systémů. Provádí audity prodejních portálů a nabízí poradenství v oblastech automatického sledování konkurence, vybraných produktů a nabízených služeb. Od roku 2015 patří mezi hlavní činnosti vývoj projektu Gloffer (Obrázek 1).

Gloffer je novinkou v online nakupování. Projekt zahrnuje webovou aplikaci, jejichž inovativní myšlenou projektu je poskytování více komplexních služeb v oblasti e-commerce na jednom místě.

V nabídce portálu Gloffer jsou tyto služby:

- Katalog produktů
- Cenový srovnávač
- Firemní prezentace
- Porovnávání produktů
- Oblíbené produkty
- Podobné produkty
- Automatizované nahrávání a párování dat e-shopů
- Propracovaný poptávkový systém
- Automatizovaný proces vystavování nabídek na poptávky
- Komunikace zákazníka s prodejcem
- Systém hodnocení prodejců a nakupujících

Gloffer se v první fázi zaměřil na český trh. V budoucnu ale plánuje rozšíření do zahraničí. Chce poskytnout svým klientům možnost jednoduché expanze. Z těchto důvodů je vyvíjen způsobem, který umožňuje jednoduše přeložit obsah webu do více jazykových mutací. Aktuálně je udržován v češtině, slovenštině a angličtině. Kromě jazykových mutací se vývoj zaměřuje na více-měnovost a regionální nastavení dle vybrané oblasti. Konkurencí Glofferu jsou srovnávače zboží, poptávkové systémy a katalogy firem.

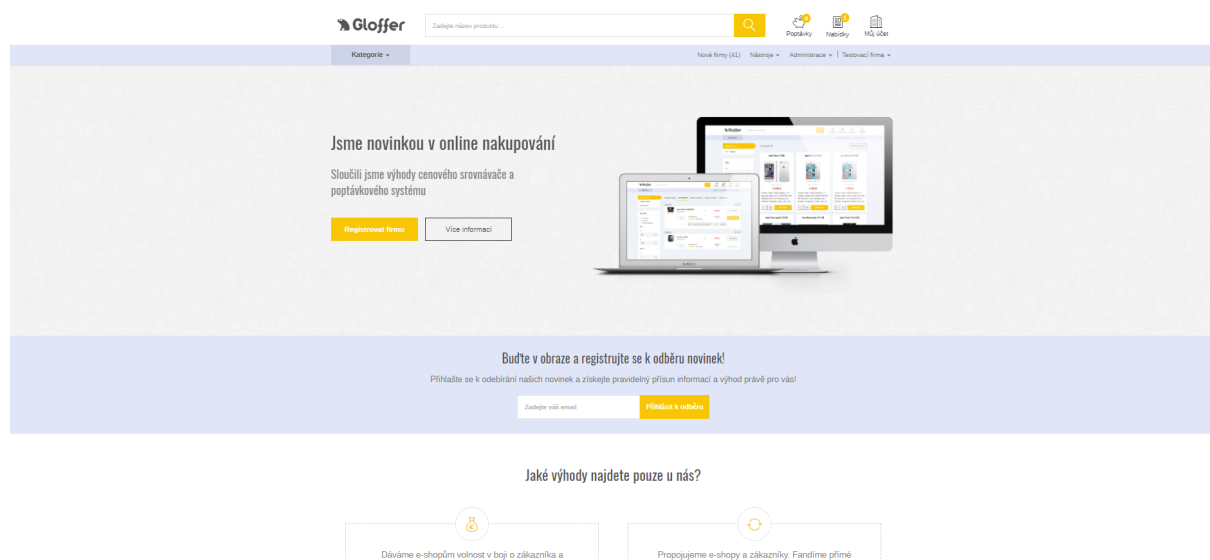
Pro splnění aktuálních trendů nabízí aplikace plně responsivní design, bez kterého by byl systém prakticky nepoužitelný na mobilních zařízeních.



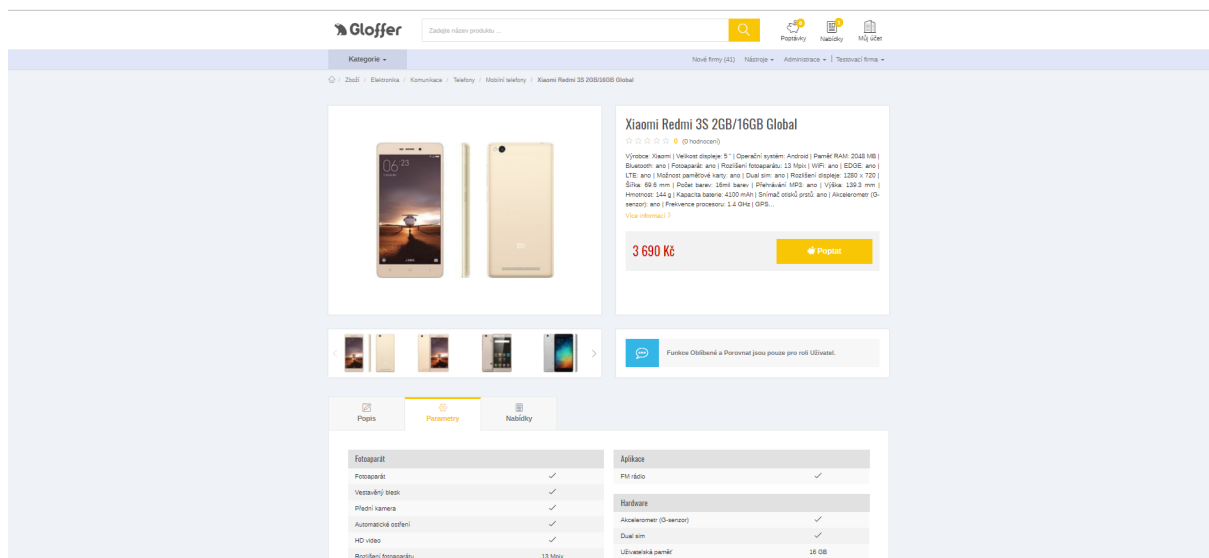
Poptávkový systém portálu se v aktuální verzi specializuje na poptávání produktů. Poskytuje dva typy poptávek. Klíčovým typem je poptávka katalogového produktu (Obrázek 2). Tuto funkcionalitu zatím neposkytuje žádný poptávkový systém v České republice. Poptávkový systém nabízí svým klientům možnost jednoduše a rychle poptat jakýkoliv katalogový produkt ze svého katalogu. Podmínkou je, aby daný produkt byl nabízen alespoň jedním prodejcem. Druhým typem je textová poptávka. Jedná se o stejnou poptávku, jakou nabízí většina existujících konkurenčních poptávkových systémů. Zadavatel poptávky zvolí kategorii, ve které se poptávané zboží nabízí, a textový popis specifikující poptávané zboží.

Poptávající má možnost vytvářet poptávky zcela zdarma. Obchodní model portálu je soustředěn pouze na nabízející firmy a je založen primárně na paušálních měsíčních platbách za přístup ke službám. Klientům, kterým tento model nevyhovuje, nabízí Gloffer férovou nabídku v podobě platby za uskutečněný obchod v podobě procenta z prodeje nebo fixní částku za transakci. Ke všem svým klientům se Gloffer snaží přistupovat individuálně a nastavit takové podmínky, které budou vyhovovat oběma stranám.

V současné době Gloffer zahajuje svou činnost a v první fázi oslovuje co největší spektrum e-shopů v České republice.



Obrázek 1: Domovská stránka portálu Gloffer



Obrázek 2: Detail katalogového produktu portálu Gloffer

## 3 Použité technologie

V této kapitole budou popsány stěžejní technologie, které budou použity při implementaci v praktické části této práce. Programovacím jazykem byl zvolen jazyk PHP verze 7.1 a framework Nette. Všechny závislosti budou instalovány pomocí technologie Composer. Pro vytváření a údržbu zdrojového kódu bude použito vývojářské prostředí PhpStorm od společnosti JetBrains. Verzování projektu zajistí verzovací systém Git. Jako základní úložiště dat byla zvolena relační databáze MySQL, která nám při správném návrhu zajišťuje integritu uložených dat. Rychlé vyhledávání a filtrace nad velkým množstvím produktů v řádech miliónů bude zajištěno za pomoci technologie Elasticsearch. Ajaxová komunikace a část systému spouštěná v prohlížeči klienta bude naprogramována v jazyce Javascript a knihovně JQuery. Pro realizaci znovupoužitelnosti, cachování a zrychlení celého systému budou sestavené dynamické filtry, parametry kategorií a výsledky vyhledávání ukládány do key-value databáze Redis. Data a informace o provedených transakcích na bankovním účtu budou získávány prostřednictvím Fio API bankovníctví Fio banky. Zakládání, provedení a potvrzování online plateb platební kartou bude zajištěno skrze GoPay API.

### 3.1 Nette

Nette je open source PHP framework pro tvorbu webových aplikací. Framework je vyvíjen organizací Nette Foundation v čele s jeho zakladatelem Davidem Grudlem.

Software využívá návrhu MVP, který se liší od více známého MVC jen v názvu řídicího prvku. Místo Controlleru se zde používá název Presenter. MVP jinak kopíruje architekturu MVC, která zajišťuje, že jednotlivé verze aplikace jsou od sebe velice dobře odděleny. Podobně jako ostatní novější PHP frameworky Nette umožňuje práci s moderními nástroji jakými jsou Dependency Injection, MVC, Web 2.0 nebo AJAX.

Při návrhu aplikací je kladen důraz na znovupoužitelnost kódu. Kvůli tomu jsou jednotlivé části rozděleny do komponent. Framework Nette také velmi dobře zajišťuje bezpečnost a obranu proti XSS, CSRF nebo útokům spojeným s odcizením session uživatele. Kódování šablon se provádí za pomoci šablonovacího systému Latte. Pro debuggování slouží ladící nástroj Tracy. Nette je velice silným nástrojem při vytváření webových formulářů, včetně live validace.

Aktuální verzi frameworku je Nette 2.4, která vyžaduje minimální verzi PHP 5.6 a je zároveň kompatibilní s nejnovější verzí PHP 7.2.[1]

### 3.2 Composer

Composer je utilita pro správu závislostí v PHP. Umožňuje projektům deklarovat závislosti knihoven a také následně tyto knihovny nainstaluje. Composer závislosti nejen získává, ale i provádí jejich aktualizaci. Pro použití Composeru je nutné mít v projektu definovaný sou-

bor *composer.json*. V něm jsou uložena meta data o projektech a knihovnách, která chceme využívat. Obvyklým nastavením je název závislosti a její verze.[2]

### 3.3 Git

Git je distribuovaný verzovací systém, který je navržen pro všechny druhy projektů. Verzovací systém slouží pro verzování aplikací. Dovoluje nahlížení do starších verzí zdrojového kódu. Je hojně využíván především při vývoji v týmech. Umožňuje programátorům pracovat na více různých verzích projektu najednou. Pro tuto funkcionalitu slouží tzv. branch. Jednotlivé verze se dají posléze spojovat, nebo slévat na stejnou úroveň.

Mezi hlavní přednosti patří vysoká efektivita a rychlost. Nejznámějšími Git repositáři je Github.com a Gitlab.com. Při vývoji jsem osobně využil službu Gitlab.com.[3]

### 3.4 MySQL

MySQL je systém řízení báze dat založený na relačním databázovém modelu. Komunikačním jazykem s touto databází je dotazovací jazyk SQL. V současné době je jejím vlastníkem společnost Sun Microsystems. Vývojářem je Oracle Corporation. MySQL databáze se řadí mezi multiplatformní. Jedná se o open source databázi podléhající licenci GPL. MySQL databáze je nejoblíbenějším databázovým systémem na světě a používají ji i přední světové aplikace jakými jsou například Youtube, PayPal, Google, Facebook, Twitter, Ebay nebo LinkedIn.[4]

### 3.5 Elasticsearch

Elasticsearch je fulltextový vyhledávač vycházející z Apache Lucene knihovny. Vyhledávač je implementován v jazyce Java, který zajišťuje multiplatformní použití. Jedná se o bezschémovou databázi, u které odpadá nutnost definovat strukturu databáze před vkládáním dat.

Struktura je jednoduše vytvořena automaticky dle vložených dat. Celá databáze je složena z jednoho nebo více clusterů, v těch se nacházejí jednotlivé uzly. V Elasticsearch jsou data uložena v indexu, ve kterém se nachází jednotlivé dokumenty. Základním datovým formátem pro uložená data a dotazy do databáze je JSON. Hlavní vlastností je rychlost a schopnost vyhledávat ve velkém množství dat v reálném čase. Mezi přednostmi patří komunikace probíhající přes REST API, vysoká dostupnost služby a snadná škálovatelnost.

Tato technologie je nejčastěji použita u webových projektů v disciplínách filtrování a vyhledávání. Pro monitorování stavu struktury databáze, nahlížení do dat, provádění a testování dotazů nad tímto systémem je možné využívat plugin Kibana.<sup>1</sup> Rozsáhlá dokumentace je dostupná na stránkách technologie.<sup>2</sup>

Elasticsearch je poskytován zdarma s dodržáním podmínek licence Apache.[5] [6]

---

<sup>1</sup>Kibana (<https://www.elastic.co/downloads/kibana>) ze dne 13.2.2018

<sup>2</sup>Elasticsearch (<https://www.elastic.co/>) ze dne 13.2.2018

### 3.6 Redis

Redis je oblíbená klíč-hodnota databáze. Často je využívána jako cache, message broker nebo jako sdílené úložiště pro různé procesy a statistiky.

V Redisu jsou pro rychlou práci data uložena v paměti RAM, což značně limituje velikost použitých dat na řádově desítky až stovky GB. Umožňuje také ukládat data perzistentního charakteru, ale u této funkcionality je nutné kalkulovat s velikostním omezením. Pro stálé uložení dat nabízí dvě cesty. První je režim RDB, kdy dochází k otisku celé RAM paměti. Druhým způsobem je AOF, při kterém je vytvářen log všech operací.

Redis podporuje klasické datové struktury jakými jsou textové řetězce, asociativní pole (hashes), seznamy, sady, seřazené množiny a bitmapy. V Redisu můžeme využívat atomické operace typu připojení řetězce, inkrementování hodnoty v asociativním poli, vkládání položek do listů, výpočty průniků, sjednocení a rozdílů nad sadami či získávání prvků ze seřazených sad.

Velkým benefitem je zabudovaná škálovatelnost. Zabezpečení přístupu k databázi je realizováno pomocí autorizačního hesla.

Při vytváření klíčů databáze je vhodné využívat zavedených konvencí:

- Používat logické členění klíčů znakem dvojtečky - Například pro data o uživateli s id 100 je vhodné použít zápis „user:100“.
- Nevytvářet příliš krátké klíče - Použití příliš krátkého klíče zhoršuje čitelnost, pokud například chceme uložit klíč „u1000flw“, je vhodnější použít čitelnější zápis „user:1000:followers“.
- Nevytváření příliš dlouhých klíčů - Důvodem je porovnávání klíčů při vyhledávání.
- Dodržení maximální velikost klíče - Výchozí velikostí je 512MB.[7]

Do dokumentace je možno nahlédnout na webových stránkách databáze.<sup>3</sup>

### 3.7 Fio API

Fio API bankovníctví slouží jako bezpečná a automatizovaná forma zpracování výpisů pohybů transakcí účtů Fio banky. Hlavními uživateli jsou e-shopy a společnosti, které potřebují stahovat informace o transakcích do svých účetních softwarů hlavně z hlediska párování platby na objednávku s následným odesláním zboží či povolením služby zákazníkovi.

Rozhraní slouží dále také pro zadávání jednotlivých a hromadných plateb. API nabízí komunikaci v datových formátech XML, CSV, JSON, HTML a PDF. Veškerá komunikace mezi klientem a bankovní institucí je zabezpečena šifrováním pomocí SSL protokolu. Aby mohl klient spravovat svůj účet přes API, musí si nejprve vygenerovat token pro komunikaci a nastavit pravidla, které akce budou povoleny a které zakázány pro daný token.

---

<sup>3</sup>Redis <https://www.redis.io/> ze dne 13.2.2018

Celá funkcionalita API je pro všechny klienty se založenými účty zdarma. Pro klienty, kteří si nedokáží nebo nechtějí vytvářet vlastní implementaci komunikace s API, je vytvořena služba Fio API Plus. Jedná se jednoduchou desktopovou aplikaci, jež umožňuje komunikaci s API. Ani tato aplikace není pro zákazníky Fio banky zpoplatněna.[8]

### 3.8 GoPay API

GoPay REST API je nástroj, prostřednictvím kterého mohou klienti komunikovat s platební bránou GoPay. API má dvě verze. První je testovací verze pro vývojáře s připravenými testovacími schémata.<sup>4</sup> Druhou je verze produkční.

U odchozí i příchozí komunikace je použito kódování UTF-8. Bezpečnost spojení zajišťuje autorizační princip OAuth2.0. Komunikace s API je rozdělena na dva druhy. Jedním je synchronní komunikace, která je využívána například při založení, zadání a zpracování online platby, kdy dochází k přesměrování na stránku GoPay a po dokončení platby zpět na web obchodníka. Dalším druhem je asynchronní komunikace, kdy jsou obchodníkem na předem definovanou adresu zasílané notifikace o stavech plateb.

V rámci zrychlení a ulehčení integrace platební brány do systému klienta je ze strany GoPay nabízena implementace pro PHP, Python, iOS, Java a Android. Další možností je nástroj příkazové řádky cURL. GoPay API je podobně jako Fio API poskytováno zákazníkům zdarma. GoPay se přizpůsobil v roce 2017 situaci české legislativy a nabízí obchodníkům možnost řešení EET při využití jedné sazby DPH na straně platební brány. V případě, že obchodník nabízí zboží ve dvou sazbách DPH, musí rozšířit komunikaci s API o předávání daňových informací pro Finanční správu.<sup>5</sup> [9]

API má kvalitní a přehlednou online dokumentaci pro každé SDK.<sup>6</sup>

---

<sup>4</sup>Provádění plateb v testovacím prostředí GoPay API (<https://help.gopay.com/cs/tema/integrace-platebni-brany/technicky-popis-integrace-platebni-brany/provadeni-plateb-v-testovacim-prostredi>) ze dne 23.2.2018

<sup>5</sup>Finanční správa (<http://www.financnisprava.cz/cs/financni-sprava/eet>) ze dne 23.2.2018

<sup>6</sup>Dokumentace GoPay API (<https://doc.gopay.com/cs>) ze dne 23.2.2018

## 4 Parametrické vyhledávání

V této kapitole se budu zabývat prvním z bodů diplomové práce. Nejprve provedu rozbor metod formulace dotazů vyhledávání s důrazem na parametrické vyhledávání v katalogových produktech a možnosti cachování dat. Následovat bude návrh a realizace vlastního řešení, které bude integrováno do portálu Gloffer. Z důvodu ověření hypotézy o cachování provedu na závěr porovnání výkonu s a bez použití předzracovaných dat.

### 4.1 Rozbor parametrického vyhledávání

V rámci analýzy aktuálního stavu v oblasti parametrického vyhledávání je zapotřebí nejprve definovat pojem katalog produktů. Následně bude provedena analýza vybraných existujících řešení na českém trhu. Dalším krokem budou metody formulace dotazů vyhledávání v rozsáhlých katalozích produktů. Vzhledem k očekávanému velkému množství dat budou probrány možnosti cachování dat.

#### 4.1.1 Katalog produktů

Pojem katalog obecně vyjadřuje uspořádaný seznam nebo výčet dle vhodných pravidel. Katalog může být reprezentován různými formami, buďto klasickou tištěnou, nebo digitální podobou. Produktový, též výrobkový katalog, je složen z jednotlivých výrobků.

Katalog produktů je složen z jednoznačně identifikovatelných výrobků. Pro identifikaci výrobků slouží EAN, což je celočíselný kód využívaný v maloobchodu i velkoobchodu. Pro EAN existuje několik standardů. Dvěma nejznámějšími je třináctimístný a osmimístný EAN. EAN je složen z prefixu, kódu výrobce, kódu výrobku a kontrolního čísla. Prefixem se označuje buď země výrobce, nebo druh výrobku a je společně s kódem výrobce přidělován společností GS1 International. Kód výrobku je pořadové číslo, které mu přiřazuje sám výrobce. Kontrolní číslo je vypočítáno ze všech předchozích čísel a slouží ke kontrole pravosti EAN kódu. EAN kód je většinou reprezentován čárovým kódem, který se skládá z čar a mezer. Šířka čar a mezer vychází z jednotlivých čísel kódu. Existují i speciální pojmenované EAN kódy, zástupcem je například kód ISBN, jenž slouží pro identifikaci knižních vydání a publikací.[10]

#### 4.1.2 Srovnání existujících řešení parametrického vyhledávání

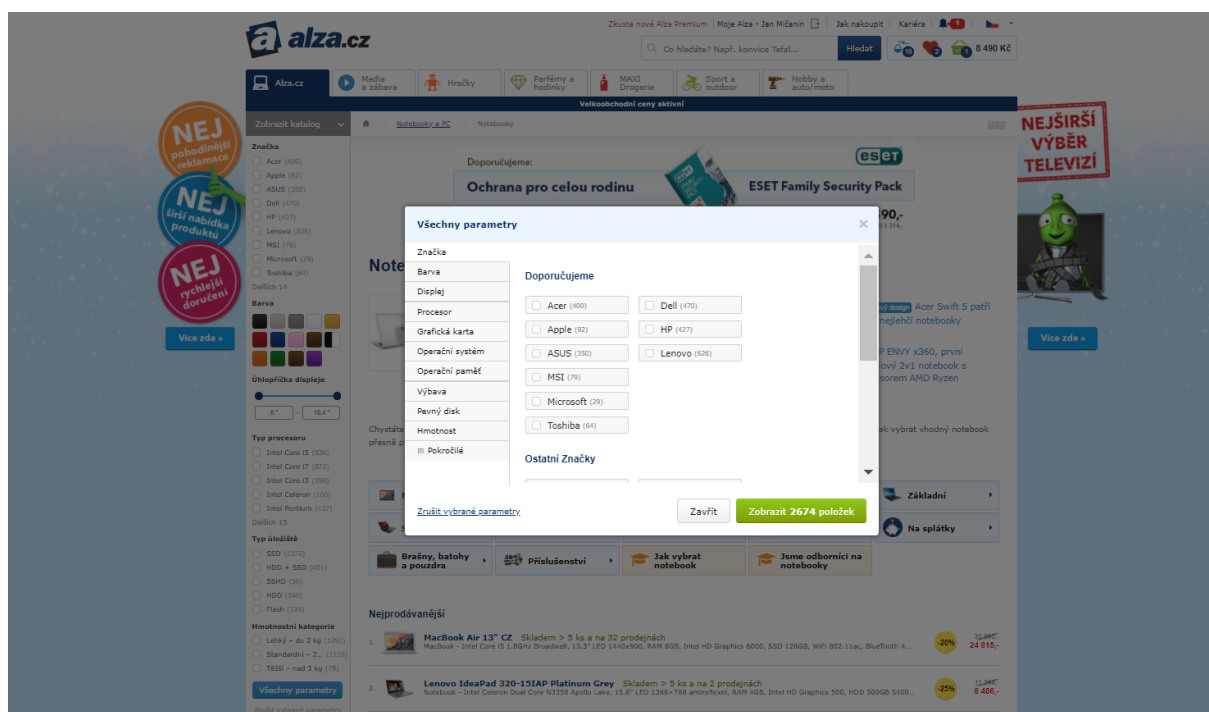
V této podkapitole se budu věnovat existujícím řešením parametrického vyhledávání. Konkrétně srovnám řešení leaderů na českém trhu, mezi které patří od e-shopy Alza.cz (Obrázek 3), CZC.cz (Obrázek 4) a nákupní galerie Heureka.cz (Obrázek 5) a srovnávač zboží Zboží.cz (Obrázek 6).

Z pohledu technologické stránky všechna testovaná řešení, výjma Zboží.cz, po každé úpravě filtru posílají ajaxový požadavek na server, který přepočítává počty zobrazených produktu a počty možných produktů v případě zvolení dalších vlastností. Výsledek vyhledávání u Alza.cz, CZC.cz a Zboží.cz je zobrazován dynamicky bez nutnosti překreslovat stránku. Heureka.cz tuto

možnost nenabízí, zde je nutné obnovení celé stránky. Z důvodu ajaxového překreslování výsledku vyhledávání ukládá Alza.cz, CZC.cz a Zboží.cz zvolené vlastnosti přímo do URL adresy. Heureka.cz vybrané parametry ponechává uložené v prohlížeči klienta do doby znovu načtení celé stránky, kdy dochází k uložení filtru do URL adresy. Stránkování produktů nabízí všechny testované systémy. Dynamické načtení další stránky s produkty nabízí jen Alza.cz a CZC.cz.

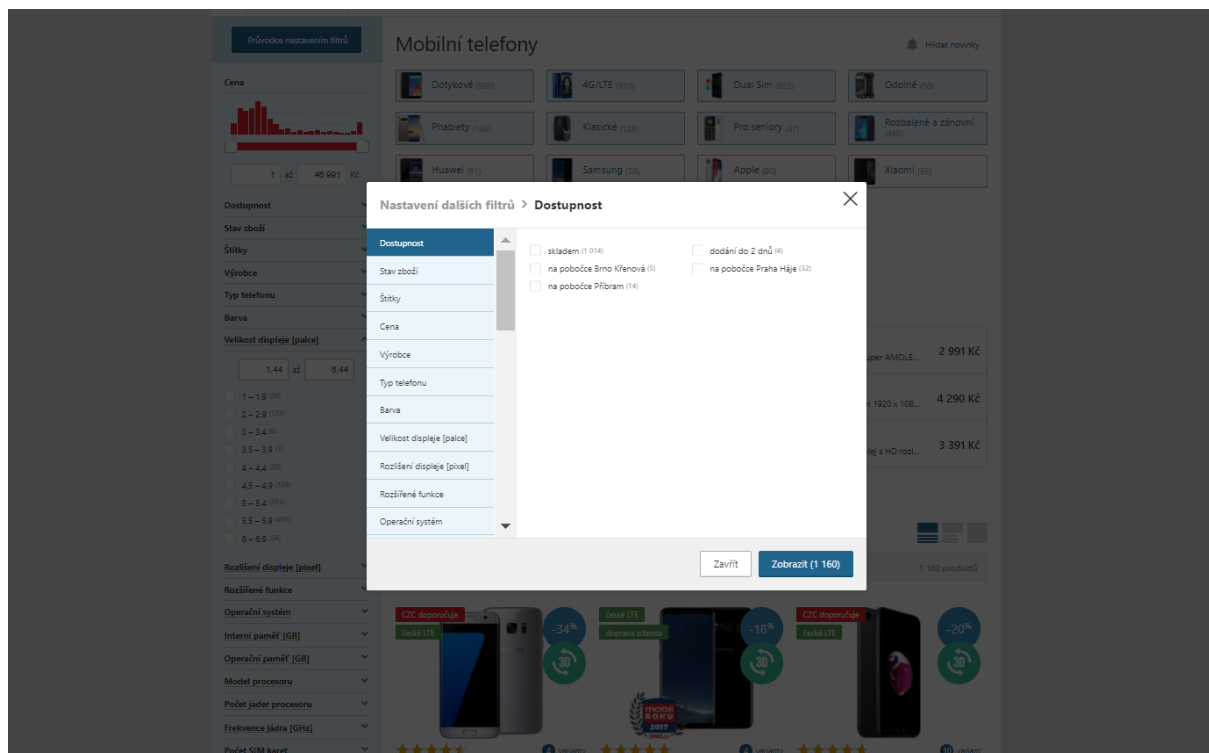
Obsah, tedy jednotlivé parametry a jejich hodnoty, je u všech testovaných systémů téměř totožný. Alza.cz (pouze v podrobném filtru) a Heureka.cz vizuálně obaluje vlastnosti do skupin.

Designově jsou si všechna řešení také velmi podobná. Filtr se u všech nachází na levé straně výsledku vyhledávání. Alza.cz a CZC.cz má v levém sloupci jen vybrané vlastnosti. Všechny parametry se ukrývají v modálním okně, které se zobrazí kliknutím na příslušné tlačítko. Ostatní mají zobrazeny všechny filtry přímo ve filtračním sloupci. Všechny testované aplikace nabízí možnost setřídít si výsledek vyhledávání.[11][12][13]



Obrázek 3: Ukázka detailního parametrického filtru vyhledávání e-shopu Alza.cz





Obrázek 4: Ukázka detailního parametrického filtru vyhledávání e-shopu CZC.cz



Bez omezení

Dostupnost

☐ Pouze skladem

K odběru v okolí

Celá ČR

Výrobce

Hledat výrobce

☐ Apple  
☒ HUAWEI  
☒ Samsung  
☒ Nokia  
☒ Xiaomi  
☐ Lenovo

Dalších 70

Počet jader procesoru

Platforma

Vlastnosti

☐ dotykový displej  
☐ fotoaparát  
☐ WiFi  
☐ LTE  
☐ dual SIM  
☐ GPS  
☐ bluetooth  
☐ svítlna

Dalších 23

Rozšířený filtr

- operační systém Android 6.0
- úhlopíčka displeje 5"
- rozlišení 1280 x 720 (HD/WXGA)
- operační paměť 3 GB

+ Další varianty

2 990 Kč - 5 499 Kč

v 94 obchodech

Porovnat ceny

Huawei P10 Lite Dual SIM, modrý

- operační systém Android 7.0
- úhlopíčka displeje 5.2"
- rozlišení 1920 x 1080 (FHD)
- operační paměť 3 GB

+ Další varianty

4 750 Kč - 7 989 Kč

v 49 obchodech

Porovnat ceny

Xiaomi Redmi Note 4 3/32 GB, černý

- operační systém Android 6.0
- úhlopíčka displeje 5.5"
- rozlišení 1920 x 1080 (FHD)
- operační paměť 3 GB

+ Další varianty

2 717 Kč - 5 389 Kč

v 45 obchodech

Porovnat ceny

Honor 9 Lite, šedý

- operační systém Android 8.0
- úhlopíčka displeje 5.65"
- rozlišení 2160 x 1080
- operační paměť 3 GB

+ Další varianty

5 401 Kč - 9 489 Kč

v 83 obchodech

Porovnat ceny

Honor 9 64 GB, Sapphire Blue

- operační systém Android 7.0
- úhlopíčka displeje 5.15"
- rozlišení 1920 x 1080 (FHD)
- operační paměť 4 GB

+ Další varianty

7 808 Kč - 13 490 Kč

v 74 obchodech

Porovnat ceny

Honor 6A 16 GB, šedý

- operační systém Android 7.0
- úhlopíčka displeje 5"

2 869 Kč - 3 690 Kč

Obrázek 6: Ukázka filtru vyhledávání cenového srovnávače Zbozi.cz

### 4.1.3 Metody formulace dotazů vyhledávání v rozsáhlých katalogích produktů

Po prostudování webových portálů, které jako svůj základ používají katalog produktů, existuje několik způsobů jakými lze vyhledávat. Mezi tyto systémy patří hlavně e-shopy nebo srovnávače produktů. Prvním způsobem nalezení výsledků je fulltextový dotaz. Dalším způsobem je vyhledávání dle produktové kategorie. Kategoricky zařazené produkty je možno následně filtrovat na základě jejich parametrů. Nejsložitější filtrační metodou je kombinace všech dříve uvedených způsobů.

**4.1.3.1 Fulltext** Fulltextové vyhledávání je technika založená na porovnání vyhledávané fráze dotazu s jednotlivými slovy v dokumentu nebo fulltextové databázi. Nejdůležitějším krokem vytváření fulltextové databáze je indexace dokumentů. V rámci fulltextu jsou využívány sady algoritmů, které dokáží u daných záznamů vytvořit statistiky výskytu frází. Výsledky vyhledávání jsou seřazovány v pořadí daném těmito statistikami.

Fulltextové vyhledávače pracují s četností výskytů klíčových slov, jejich umístěním v rámci textu, skloňováním, časováním či kořeny slov. Rozdíly mezi jednotlivými vyhledávači je možné sledovat v kvalitě zpracování překlepů. U katalogových produktů se primárně vyhledává dle názvu, výrobce a popisu.

Při potřebě nasazení fulltextového vyhledávání do vlastního systému není obvykle zapotřebí vymýšlet a programovat vlastní algoritmy. Pro malý systém je možné využít fulltextových indexů a dotazů v klasických relačních databázích. Pro práci s daty v řádech GB není vhodné použít relační databázi pro fulltextové vyhledávání. Pro tyto případy existují fulltextové databáze, které je možné si buď za poplatek, nebo zdarma nainstalovat na vlastní servery. Mezi zástupce patří například Elasticsearch 3.5 nebo Sorl. Alternativou je využití služeb typu Algolia, kdy jsou data uložena na serverech třetích stran a kdy dotazování probíhá přes API. Fulltextové databáze jsou při správné konfiguraci a dostatečném hardware velmi rychlé, konkrétně Algolia uvádí dobu čekání relevantních výsledků pod 100ms kdekoli na světě.

Fulltextové vyhledávání nejvíce využívají internetové vyhledávače, jakými jsou např. Google.com, Bing.com nebo Yahoo.com. V České republice mezi zástupce patří Seznam.cz nebo Centrum.cz. Kvalita eshopových vyhledávačů se nedá srovnávat s internetovými. Internetové jsou mnohem sofistikovanější a složitější. Zpracovávají nepoměrně větší množství dat. Mají vlastní roboty, tedy skripty, jejichž úkolem je procházet jednotlivé webové stránky a hodnotit je na základě různých parametrů kvality, důležitosti a důvěryhodnosti. Tyto parametry následně určují pořadí zobrazení ve výsledku vyhledávání. Mezi nejznámější fulltextové algoritmy patří PageRank od společnosti Google.[15][16]

**4.1.3.2 Parametrické vyhledávání** Parametrické vyhledávání se zabývá vyhledáváním nad jednotlivými parametry produktů. Prakticky každý produkt, který existuje, má své parametry. Mezi základní parametry patří EAN výrobku, název, výrobce a cena. Dalšími parametry závisí na druhu produktu. Může se jednat například o rozměr, rychlost, vodotěsnost, kompatibilitu atd.

Pro účely vyhledávání rozdělujeme vlastnosti produktů do několika skupin:

- Výčtový parametr – Je takový parametr, kde máme na výběr z konečné množiny hodnot. Příkladem může být výrobce velikost RAM paměti atd.
- Rozsahový parametr - Patří zde vlastnosti, které lze filtrovat rozsahem hodnot od a do. Typickými zástupci jsou cena produktu, šířka, výška, hmotnost nebo kapacita baterie.
- Parametr typu ano/ne – Jedná se o parametr, který říká, zdali danou vlastnost produkt má či ne. Mohou nastat tedy jenom dvě možnosti. Pro příklad u mobilních telefonů můžeme jasně definovat, zdali daný telefon má Bluetooth, WiFi, NFC, snímač otisků prstů, gyroskopický senzor atd.

Při definování parametrů jsou dvě možnosti. Buď přidáním hodnot vlastností spektrum potencionálních výsledků rozšiřujeme nebo omezujeme. Pro rozšíření jsou obvykle využity výčtové parametry. Naopak rozsahové a parametry typu ano/ne nám výsledek vyhledávání omezují.

Kategorické vyhledávání je specializací vyhledávání parametrického. Kategorie je v tomto případě typ výčtového parametru, který nabývá právě jedné hodnoty. Při vyhledávání v kate-

goriích se obvykle využívá stromu kategorií. To znamená, že jednotlivé kategorie jsou do sebe hierarchicky zařazeny. Velmi častým jevem u existujících řešení bývá, že při zobrazení nadřazené kategorie se zobrazují ve výsledku produkty z dané kategorie a podkategorií všech úrovní.

**4.1.3.3 Kombinace fulltextového dotazu a parametrického vyhledávání** Princip kombinace fulltextového dotazu a parametrické filtrace spočívá v tom, že nejprve jsou vyfiltrovány produkty na základě parametrů. Následně je nad zmenšenou množinou proveden fulltextový dotaz. Důvod této posloupnosti je ten, že parametrické vyhledávání je méně náročné na výpočetní výkon. V opačném případě by docházelo ke zbytečnému zpomalení vykonávání dotazu.

#### 4.1.4 Cachování

Pojem cache v IT znamená softwarovou nebo hardwarovou komponentu, která slouží jako mezipaměť k uchovávání často používaných dat a výpočetně složitých dat. Pro paměť cache se využívají rychlé a drahé paměti, jakými jsou například paměti RAM.

Cachování dat se obecně definuje jako uschování nebo předzpracování dat pro další použití. Jedná se o metodu, která razantním způsobem zvyšuje rychlost webových aplikací a snižuje hardwarovou náročnost celé aplikace. Nevýhoda cachování může nastat v případě, že se při nevhodných aktualizacích a zároveň nevhodné správě paměti budou zobrazovat zastaralé předzpracované informace. Z toho důvodu by data uložená v cache měla mít nastavenou expiraci, aby nedocházelo k zaplňování cache již nepoužívanými daty a dlouhodobému zobrazování starých informací.

Cachování obsahu má význam u všech webových projektů. U těch rozsáhlejších již v dnešní době není možné poskytovat bez cachování kvalitní a rychlé služby vzhledem k velkému počtu uživatelských požadavků.

U webových stránek se využívají nejméně dva různé typy cachování.

**4.1.4.1 Cache webového prohlížeče** Jedná se o cache vytvořené na straně internetového prohlížeče. Ten také zajišťuje její správu. Data jsou uložena přímo v paměti klienta. Uživatel prohlížeče si může tento způsob cachování zakázat. Do této cache se primárně ukládají opakovaně načítané soubory jakými jsou kaskádové styly, javascripty, webové fonty nebo obrázky. Cache prohlížeče šetří cenné milisekundy při načítání stránky a stovky KB až jednotky MB přenášených dat, což zvyšuje rychlost vykreslování stránek. Co se přenášených dat týče, v dnešní době rychlého internetu se jedná hlavně o data na mobilních zařízeních s připojením na mobilní internet, kde jsou stále využívány omezení typu FUP.

**4.1.4.2 Cache serveru** Cachování probíhá na straně serveru. Jedná se o uložení sestavených dat, která následně při požadavku klienta server odesílá zpět v odpovědi. Takto předpřipravená data bývají serializovaná. Serializace dat znamená převedení složitých datových struktur jakými

jsou objekty nebo pole do textové podoby, kterou je možné jednoduše zapsat jako prostý text do souboru. Opakem serializace je deserializace, tedy sestavení objektů z textové podoby.

Princip cachování webové aplikace na serveru je následující. Požadovaná data jsou při prvním požadavku sestavena z persistentního úložiště a následně uložena do cache. Při dalším načtení nahlédne aplikace do cache, a pokud se v ní nacházejí požadovaná data, tak jsou vrácena. V ten moment dochází k úspoře výpočetního výkonu serveru.

Neprimitivnějším způsobem implementace cache na straně serveru je uložení v souboru ve složce na pevném disku. Dalším, více využívaným způsobem, je využití cachovací NoSQL databáze.

Pro cache se nejčastěji využívají databáze, ve kterých jsou data ukládána v paměti RAM. Vzhledem k ceně RAM si musí programátor dobře rozmyslet, pro která data bude tuto cache využívat. Tato databáze bývá nainstalována na stejném stroji jako webová aplikace nebo na stroji, který je v blízkosti serveru s webovou aplikací. V tomto případě jsou oba počítače propojeny vysokorychlostní sítí, aby časový dopad přenášení dat z databáze na webový server byl co nejmenší. Mezi nejpopulárnější zástupce patří Redis a Memcached.

Výhodou použití key-value databáze oproti klasickému souboru na disku přímo ve webové aplikaci patří škálovatelnost, která způsobuje zvyšování dostupnosti a rychlosti key-value databáze. Další výhodou oproti klasické souborové cache je ta skutečnost, že k databázi může přistupovat více než jedna instance webového serveru. To vede také k deduplikaci cachovaných dat a snížení paměťové náročnosti projektu.

Rychlost a dostupnost prvního načtení webových aplikací je možné zvýšit za pomoci získaných statistik o prohlížených stránkách a obsahu. V tomto případě je vhodné na jejich základě obsah v pravidelných intervalech předzpracovávat. Proces lze realizovat například pomocí nástroje Cron, který spouští zpracování frekventovaných dat a jejich uložení do cache. Cron je automatizovaný serverový softwarový nástroj spouštějící příkazy nebo skripty v definovaném čase.

**4.1.4.3 Porovnání Redis a Memcached** Memcached byl originálně navržen jako cachovací úložiště. Redis byl navržen primárně jako NoSQL databáze, cachovací úložiště a message broker. V tabulce 2 se nachází srovnání in-memory key-value úložiště Memcached a Redis.

Na základě srovnání je zřejmé, že Redis poskytuje více funkcionalit. Konkrétně se jedná o podporu typovosti, replikování, distribuci dat, podporu uložených procedur, transakcí a persistentní uložení dat. Oproti Memcached podporuje také více programovacích jazyků.

Na obrázku 7<sup>7</sup> je znázorněno hodnocení DB-Engines Ranking popularity obou technologií. Z hodnocení lze vyčíst, že Redis je aktuálně 4x populárnější než Memcached. [17]

---

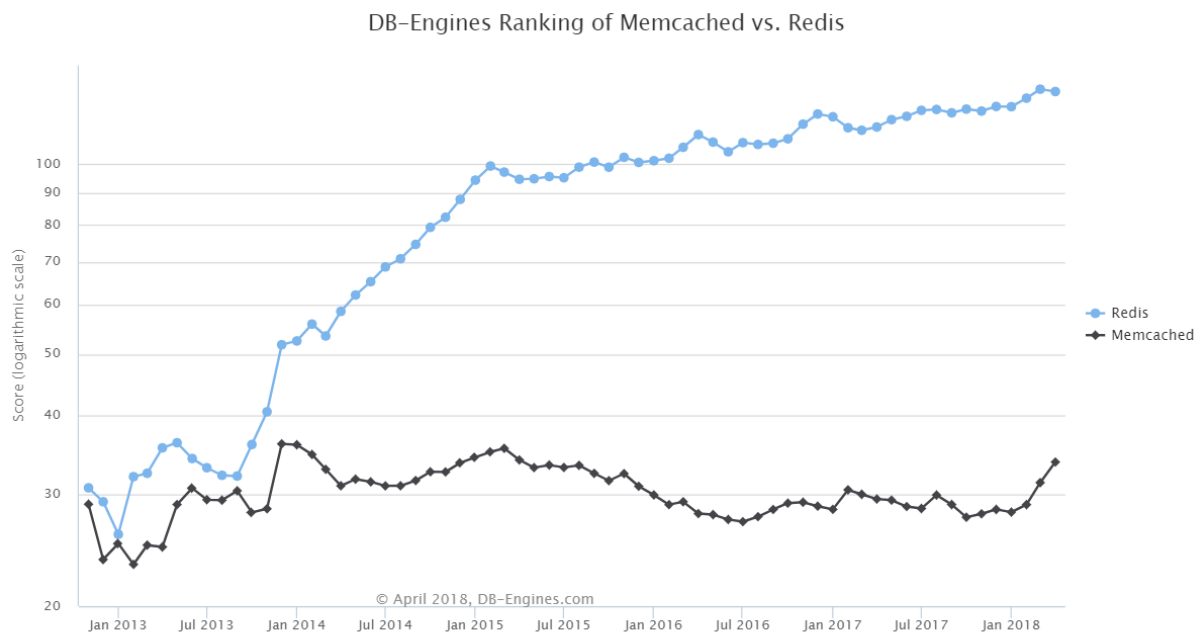
<sup>7</sup>Graf hodnocení popularity ([https://db-engines.com/en/ranking\\_trend/system/Memcached;Redis](https://db-engines.com/en/ranking_trend/system/Memcached;Redis)) ze dne 1.4.2018

Tabulka 2: Srovnání Memcached a Redis

Název	Memcached	Redis
Primární databázový model	Key – value úložiště	Key – value úložiště
Sekundární databázový model		Document store Graph DBMS Time Series DBMS
Webová stránka	<a href="http://www.memcached.org">www.memcached.org</a>	<a href="http://redis.io">redis.io</a>
Dokumentace	<a href="https://github.com/memcached/memcached">github.com/-memcached</a>	<a href="http://redis.io/documentation">redis.io/documentation</a>
Vývojář	Danga Interactive	Salvatore Sanfilippo
První spuštění	2003	2009
Aktuální verze	1.5.6, Únor 2018	4.0.9, Březen 2018
Licence	Open Source	Open Source
Pouze jako cloudová služba	ne	ne
Implementační jazyk	C	C
Operační systém serveru	FreeBSD Linux OS X Unix Windows	BSD Linux OS X Windows
Databázové schéma	bez schématu	bez schématu
Typovost	ne	částečně
Sekundární indexy	ne	ne
SQL podpora	ne	ne
Přístup skrze API	vlastní protokol	vlastní protokol
Podporované jazyky	.Net C C++ ColdFusion Erlang Java Lisp Lua OCaml Perl PHP Python Ruby	C C# C++ Clojure Crystal D Dart Elixir Erlang Fancy Go Haskell Haxe Java

		JavaScript (Node.js)
		Lisp
		Lua
		MatLab
		Objective-C
		OCaml
		Pascal
		Perl
		PHP
		Prolog
		Pure Data
		Python
		R
		Rebol
		Ruby
		Rust
		Scala
		Scheme
		Smalltalk
		Swift
		Tcl
		Visual Basic
Uložené procedury	ne	Lua
Triggery	ne	ne
Distribuce dat	není	Sharding
Replikace	není	Master-slave relikace
		Multi-master replikace
MapReduce	ne	ne
Cizí klíče	ne	ne
Transakční koncepty	ne	Optimistic locking
		atomické provádění operací
Persistentní uložení dat	ne	ano
Zabezpečení připojení	SASL	Free verze - heslo
		Komerční verze – SSL, ACL





Obrázek 7: Trend vývoje popularity technologií Memcached a Redis

## 4.2 Návrh a realizace parametrického vyhledávání

Při návrhu jsem využil primárně informace a zkušenosti získané analýzou existujících řešení na trhu.

Datové podklady pro parametrické vyhledávání poskytla rozsáhlá databáze portálu Gloffer obsahující kategorie, produkty, jejich vlastnosti a hodnot vlastností. Bez této databáze by nebylo možné vyhledávání realizovat a testovat.

Kapitola je rozdělena do několika částí. V první se budu věnovat návrhu databázové struktury modulu. Dále popíšu algoritmus vygenerování datových podkladů pro samotný dynamický filtr, tedy sestavení kolekce vlastností a jejich hodnot dané kategorie. Na základě těchto vlastností bude následovat sekce věnovaná implementaci komponenty sloužící k vytvoření dynamického filtru ve frontendovém sektoru portálu Gloffer. Sestavená data pro parametrická vyhledávání je možné využít i v jiných částech systému. Těmto částem je věnována následující kapitola. Následovat bude vizualizace vyhledávání. Další kapitolou bude proces sestavení dotazu parametrického vyhledávání a vrácení výsledků. Z důvodu znovupoužitelnosti výsledků se budu věnovat také cachování dat.

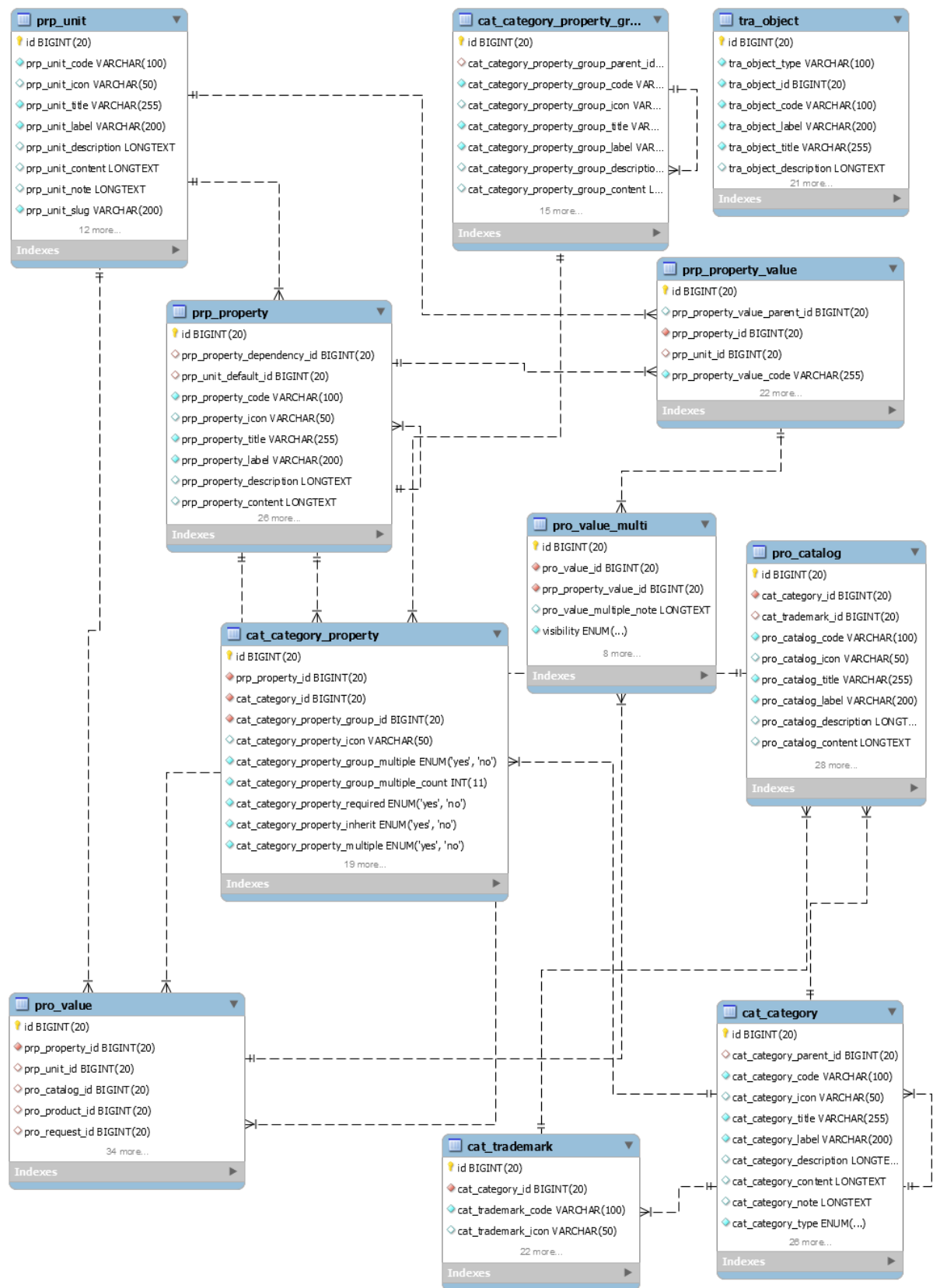
Základním úložištěm pro data persistentního charakteru je relační databáze MySQL. Pro účely rychlého vyhledávání byla tato data zkopírována do vyhledávacího systému Elasticsearch. Pro cachování dat jsem využil jednu z nejpopulárnějších cachovacích databází Redis. Modul kategorií produktů a dynamických filtrů je navrhován a implementován s ohledem na veškeré jazykové mutace systému Gloffer.

#### 4.2.1 Databázový návrh

Při návrhu databázové struktury (Obrázek 8) jsem zohledňoval všechny potřebné části katalogu, produktů a vlastností.

Seznam použitých tabulek:

- pro\_catalog – Katalogový produkt
- prp\_property – Vlastnost produktu
- prp\_property\_value – Hodnota vlastnosti produktu
- prp\_unit – Jednotka vlastnosti
- cat\_trademark – Značka produktu
- cat\_category – Kategorie
- cat\_category\_property\_group – Skupina vlastností
- cat\_category\_property - Zařazení vlastnosti do kategorie a skupiny
- pro\_value – Hodnota vlastnosti katalogového produktu
- pro\_value\_multi – Hodnoty vlastnosti katalogového produktu
- tra\_object – Překlady všech entit v databázi



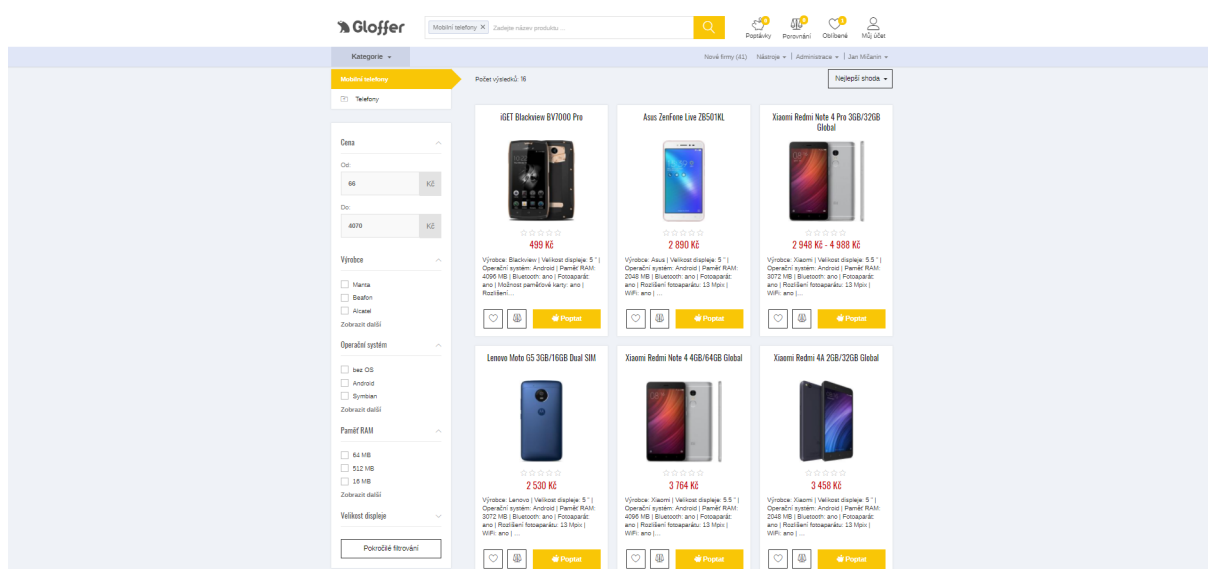
Obrázek 8: ERD model parametrického vyhledávání a kategorie produktů

#### 4.2.2 Vizuální návrh

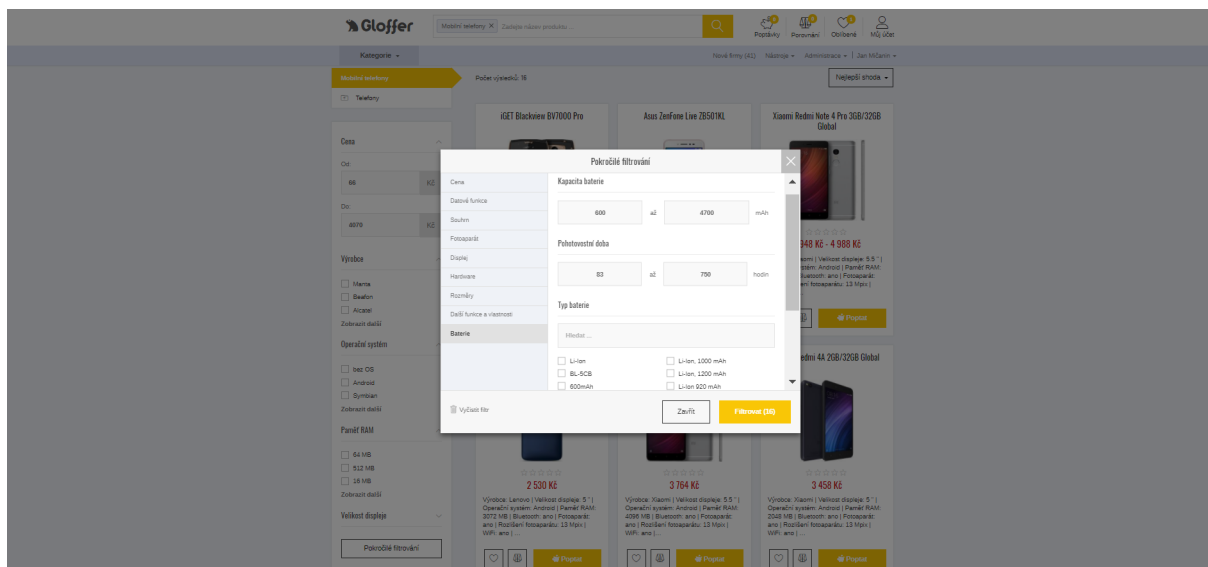
Vyhledávání vizuálně dělí na dvě části. První je tzv. Rychlý filtr (Obrázek 9). Jedná se o část je část filtrování, ve které se nachází vybrané parametry definované v databázové struktuře. V této části se pro všechny kategorie výhradně zobrazuje filtrace dle rozsahu ceny. Pro příklad v kategorii *Mobilní telefony* jsou vizualizovány vlastnosti Výrobce, Operační systém, Paměť RAM a velikost displeje. Druhou částí je tzv. Pokročilý filtr. Ten obsahuje všechny vlastnosti dané kategorie. Je zde také zkopírován obsah Rychlého filtru.

Z vizuálního pohledu je Rychlý filtr umístěn v levé části okna. Pokročilý filtr (Obrázek 10) je možné vyvolat kliknutím na tlačítko *Pokročilé filtrování*. Po kliknutí je filtr zobrazen v modálním okně. Toto okno je rozděleno na dvě části. V levé pólce se nachází skupiny vlastností. V pravé části jsou vizualizovány vlastnosti a jejich hodnoty. Ve spodní části okna jsou akční tlačítka pro vyčištění filtru, zavření okna a tlačítko na zobrazení výsledku s uvedeným počtem produktů. Toto tlačítko je také lokalizováno pod Rychlým filtrem.

Rozsahové vlastnosti jsou definovány dvěma formulářovými textbox prvky, vlastnost typu ano/ne je vizualizována checkboxem. Pro výčtové vlastnosti je použit checkboxlist. Pokud je počet hodnot ve výčtu vysoký, je nad seznam vlastností umístěno vyhledávací okno pro hodnoty.



Obrázek 9: Ukázka rychlého filtru parametrického vyhledávání

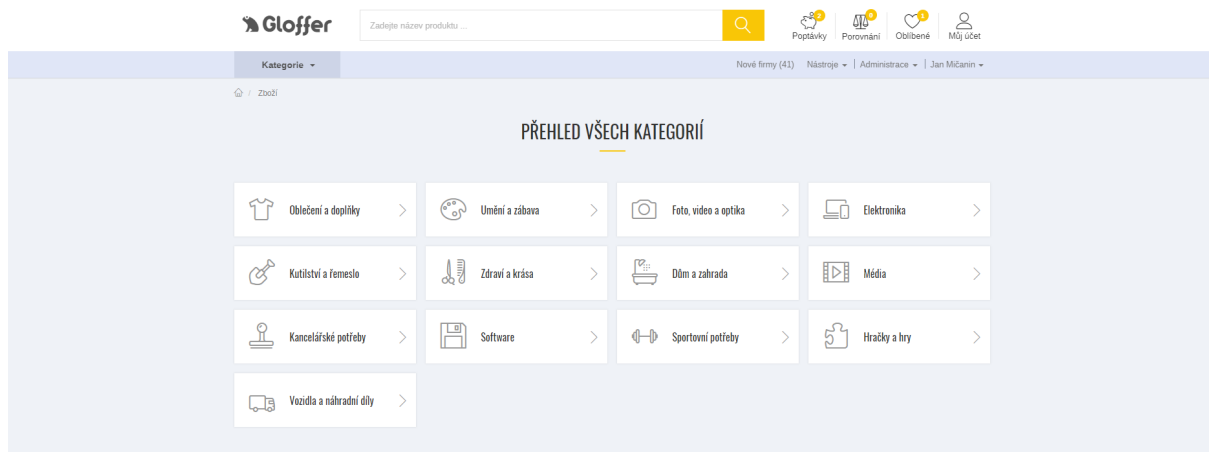


Obrázek 10: Ukázka detailního filtru parametrického vyhledávání

### 4.2.3 Kategorie produktů

Parametrické vyhledávání je založeno na všech vlastnostech produktů a jejich hodnotách. Vzhledem k faktu, že produkty jsou zařazeny do kategorií, je důležité správně navrhnout jejich strukturu. Vhodným způsobem uložení kategorií je struktura stromová. Každá obsahuje atribut s odkazem na předka. Kořenem použitých dat je *Zboží*. V celém systému se nachází přes 8 tisíc kategorií. Samozřejmostí je také jejich překlad do několika světových jazyků. Každá kategorie má řečeno, zdali je možné do ní přiřazovat vlastnosti nebo ne. Ty jsou definovány v propojovací tabulce *cat\_category\_property*. Dále se zde nachází skupina, do které je vlastnost v rámci kategorie zařazena. Veškeré kategorie dědí vlastnosti jejich předků, a tudíž není potřeba opakovaně definovat stejné vlastnosti v různých úrovních zanoření.

V rámci vytvořeného vyhledávání je možné procházet celou stromovou strukturu kategorií. Zobrazovány jsou pouze ty kategorie, do kterých jsou zařazeny produkty (Obrázek 11). Pro strom kategorií byly navrženy a implementovány vyhledávací dotazy, které slouží k získávání počtů produktů přímých potomků této kategorie. Tato data umožňují zrychlení orientace v kategoriích a následně nalezení vyhledávaného produktu. Více informací k dotazování se nachází v podkapitole Dotazování do Elasticu 4.2.6.



Obrázek 11: Kořenové kategorie produktů portálu Gloffer

#### 4.2.4 Datové podklady pro parametrický filtr

Součástí podkladů jsou data kategorií, produktů a jejich vlastností. Vlastnosti kategorií jsou definovány prostřednictvím katalogových produktů. Existuje několik druhů vlastností. Jednotlivé druhy se nachází v teoretické části této práce 4.1.3.2. Výčtové vlastnosti mají číselníky hodnot, kterých může daná vlastnost nabývat.

Sestavené datové podklady jsou uloženy do asociativního pole. Ukázka struktury dat je uvedena na výpisu 1. Z důvodu omezení paměťové náročnosti celého řešení nejsou do struktury ukládány celé ORM objekty, ale jen specifické atributy, které jsou nezbytné pro vizualizaci filtrů.

```
{
  "3704": {
    "property": {
      "46609": {
        "id": 46609,
        "show": "autocomplete",
        "show_new": "autocomplete",
        "translate": {
          "cs": {
            "label": "Vrobce",
            "title": "Vrobce",
            "description": "Vrobce",
            "content": null
          },
        },
        "en": {
          "label": "Producer",
          "title": "Producer",
```

```

        "description": "Producer\r",
        "content": null
    },
    "unit_id": null,
    "values": {
        "2274086": {
            "translate": {
                "cs": {
                    "label": "lg",
                    "title": "LG",
                    "description": "LG",
                    "content": null
                }
            },
            "unit_id": null
        },
        "2274094": {
            "translate": {
                "cs": {
                    "label": "sencor",
                    "title": "Sencor",
                    "description": "Sencor",
                    "content": null
                }
            },
            "unit_id": null
        }
    }
}

```

---

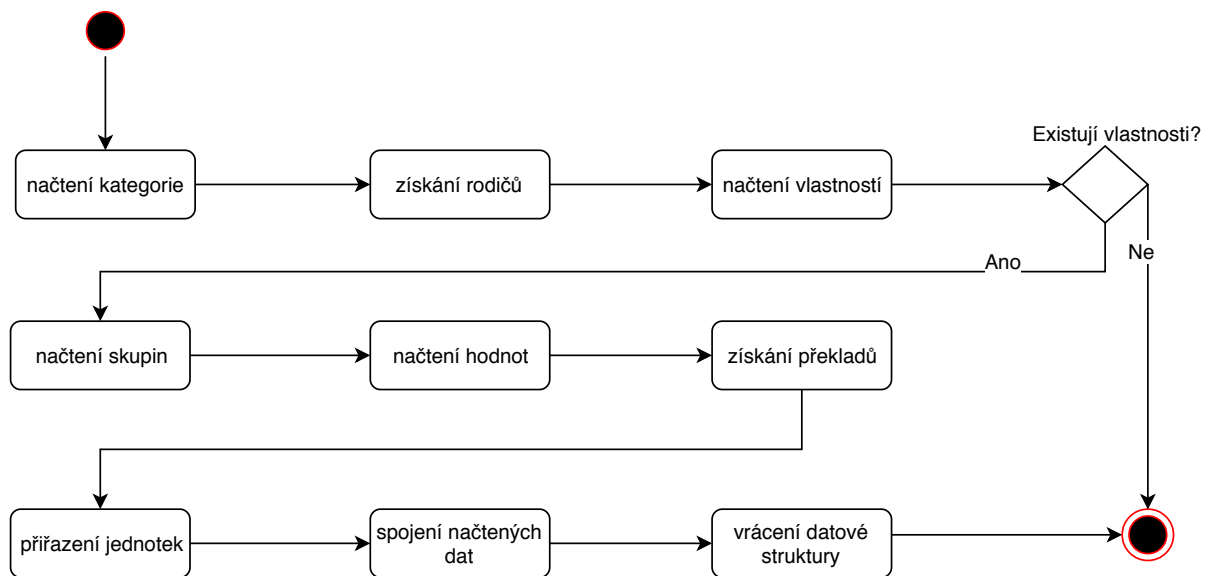
Výpis 1: Ukázka struktury dat podkladů vyhledávání

Na nejvyšší úrovni pole jsou uloženy skupiny vlastností. U těch je evidováno pole vlastností *properties*, ikona, seznam vlastností, které se nachází v rychlém filtru (pojem rychlý filtr je vysvětlen v kapitole 4.2.2), a sekce *translate* s přeloženým názvem a popisem skupiny. V seznamu vlastností se nachází jednotlivé vlastnosti. Každá vlastnost má zde sekci *values*, ve které se

nachází hodnoty dané vlastnosti, id, atribut *show* určující, jak se má vlastnost vizualizovat a sekce *translate* shodná se sekci u skupin. Pro hodnoty vlastností je evidována sekce *translate*, která je obsahově shodná s ostatními. Jako index skupin, vlastností i jejich hodnot je použit databázový identifikátor, pro jednoznačné určení, o jakou skupinu, vlastnost nebo hodnotu se jedná.

Proces přípravy datových podkladů pro parametrický filtr je definován diagramem 12 a je následující:

1. Získání instance aktuální kategorie
2. Rekurzivní sestavení seznamu všech rodičovských kategorií
3. Dotazování vlastností na základě seznamu kategorií
4. Načtení všech skupin vlastností a uložení do sestavené struktury
5. Načtení hodnot vlastností
6. Přidání překladů skupin a vlastností do struktury
7. Přiřazení jednotek k jednotlivým vlastnostem
8. Sestavení všech dat výše načtených do jedné struktury včetně jejich překladů
9. Vracení vytvořené datové struktury



Obrázek 12: Proces přípravy datových podkladů parametrického vyhledávání

Řídícím prvkem celého procesu sestavení dat je třída *CategoryPropertyManager*.



#### 4.2.5 Další využití sestavených dat

Data sestavená pro parametrické vyhledávání jsou také využita v dalších funkcionalitách systému Gloffer. Aktuálně jsou tato data ve výpisu parametrů katalogových produktů, který je umístěn jejich detailu. Dalším místem je porovnávání zboží. V této komponentě dochází ke sjednocení vlastností všech kategorií, do kterých jsou zařazeny porovnávané produkty. Poslední komponentu tvoří formulář pro vytvoření katalogového produktu.

#### 4.2.6 Dotazování a výsledek vyhledávání

Při sestavování dotazu jsem se nejprve seznámil a prostudoval způsoby vytváření dotazů databáze Elasticsearch. V rámci analýzy existujících řešení bylo také nutné nastudovat a využít technologii AJAX. Ta byla použita v kombinaci s klasickým požadavkem, při kterém dochází ke znovunačtení stránky.

Pro komunikaci s Elasticsearch jsem využil knihovnu *Elastica*<sup>8</sup> instalovanou, podobně jako ostatní použité knihovny, skrze technologii Composer.(Výpis 2)

---

```
public function getCount($query, $categoryCode, $filterConfigArray)
{
    $search = new \Elastica\Search($this->client);
    $search->addIndex($this->getElasticIndex())
        ->addType(self::DOCUMENT_TYPE);

    $path = self::INDEX_NAME . '/' . self::DOCUMENT_TYPE . '/_count';

    $queryArray = $this->getCountQuery($query, $categoryCode,
        $filterConfigArray);

    $this->query = json_encode($queryArray, JSON_UNESCAPED_UNICODE);
    $this->request($path, \Elastica\Request::GET, $this->query);
    $data = $this->getData();
    $this->responseData = isset($data["count"]) ? $data["count"] : NULL;

    return $this->responseData;
}
```

---

Výpis 2: Ukázka metody pro odeslání dotazu pomocí knihovny Elastica

Vyhledávání se skládá ze dvou základních typů dotazu. Prvním je dotaz na počet výsledků (Výpis 3). Tento typ slouží k získání počtu výsledku vyhledávání. Počet je následně vizualizován u tlačítka Zobrazit. Podobný dotaz jsem použil pro výpočet počtu produktů potomků aktuálně

---

<sup>8</sup>Elastic (<https://github.com/rufin/Elastica>) ze dne 13.3.2018

vybrané kategorie. Druhým typem je dotaz, jehož odpovědí jsou dokumenty reprezentující výsledek vyhledávání obsahující také dotaz na výchozí hodnoty rozsahových parametrů (Výpis 4). Vracený výsledek vyhledávání (Výpis 5) je následně předán do šablony a vykreslen.

---

`GET gloffer_data/pro_catalog/_count`

```
{
  "query": {
    "function_score": {
      "query": {
        "bool": {
          "filter": [
            {
              "nested": {
                "path": "category",
                "query": {
                  "bool": {
                    "filter": [
                      {
                        "term": {
                          "category.id": "1590"
                        }
                      ]
                    }
                  }
                }
              }
            ]
          }
        }
      },
      [
        {
          "nested": {
            "path": "property",
            "query": {
              "bool": {
                "filter": [
                  {
                    "term": {
                      "property.property_id": 46611
                    }
                  ]
                }
              },
              {

```

```

        "nested": {
          "path": "property.multi",
          "query": {
            "bool": {
              "filter": {
                "terms": {
                  "property.multi.property_value_parent_id": [
                    "2274016"
                  ]
                }
              },
              "should": []
            }
          }
        },
        "range": {
          "pricing.price_min_unified": {
            "gte": 62,
            "lte": 7420
          }
        }
      ],
      "functions": [
        ]
    }
  }
}

```

---

Výpis 3: Ukázka dotazu na počet výsledků vyhledávání

---

GET gloffer\_data/pro\_catalog/\_search

```
{  
  
  "aggs": {  
    "default": {  
      "aggs": {  
        "values": {  
          "aggs": {  
            "46616": {  
              "aggs": {  
                "max": {  
                  "max": {  
                    "field": "property.multi.value_number"  
                  }  
                },  
                "min": {  
                  "min": {  
                    "field": "property.multi.value_number"  
                  }  
                }  
              },  
            },  
            "filter": {  
              "term": {  
                "property.multi.property_id": 46616  
              }  
            }  
          }  
        },  
        "nested": {  
          "path": "property.multi"  
        }  
      }  
    },  
    "nested": {  
      "path": "property"  
    }  
  }  
}
```

```

    }
  },
  "price_max": {
    "max": {
      "field": "pricing.price_min_unified"
    }
  },
  "price_min": {
    "min": {
      "field": "pricing.price_min_unified"
    }
  }
},
"from": 0,
"size": 24,
"sort": {
  "id": {
    "order": "desc"
  }
}
}

```

---

Výpis 4: Ukázka dotazu na výchozí hodnoty parametrů

---

```

{
  "took": 21,
  "timed_out": false,
  "_shards": {
    "total": 4,
    "successful": 4,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 5658,
    "max_score": null,
    "hits": [
      {
        "_index": "gloffer_data_10",
        "_type": "pro_catalog",

```

```

    "_id": "97115957",
    "_score": null,
    "_source": {
      "country": "cz",
      "code": "533479152",
      "icon": null,
      "destination": "international",
      "language": "cs",
      "source": "pro_catalog",
      "translate": {
        "cs": {
          "category_path": " Zbo  | Elektronika | Komunikace | Telefony |
            Mobiln telefony",
          "category_destination": "Mobiln telefony"
          "label": "UMIDIGI C Note 3/32 GB",
          "slug": "97115957"
        }
      },
      "category_priority_level": 200,
      "ean": null,
      "priority_commercial": 0,
      "category_id": 1590,
      "stats": {
        "inspection_year": 0,
        "inspection_month": 0,
        "inspection_full": 0
      },
      "property": [
        {
          "category_property_group_id": null,
          "code": "param-name-17220",
          "icon": null,
          "show": "checkboxlist",
          "type": "number",
          "show_new": "checkboxlist",
          "category_property_group_multiple_order": 0,
          "property_id": 46637,
          "value_single_id": null,
          "multi": [

```

```

    {
      "value_icon": null,
      "value_type": "value",
      "value_title": "3072",
      "property_value_parent_id": 2275635,
      "value_order": 1000,
      "language": "cs",
      "value_content": null,
      "property_id": 46637,
      "unit_id": 3668
    }
  ],
  "dependency_id": null,
  "property_multiple_order": 0,
  "unit_default_id": null,
  "value_list_tag": "no",
  "unit_code": null,
  "id": 113216706,
  "slug": "22866818_Pamet_RAM"
}
],
"duration_end": null,
"priority_internal": 0,
"id": 97115957,
"slug": "97115957",
"duration_begin": "2017-10-13 11:05:39",
"image": [
  {
    "path": "/107/169/cf0272e088159b7acae962cf7085d0b8825ed54f.jpg",
  }
],
"visibility": "invisible",
"created": "2017-10-13 11:05:39",
"active": "enabled",
"trademark_id": null,
"category": [
  {
    "category_priority_level": 100,
    "category_priority_commercial": 0,
  }
]

```

```

        "code": "goods",
      },
      {
        "category_priority_level": 100,
        "category_priority_commercial": 0,
        "code": "goods_222"
      }
    ],
    "pricing": {
      "currency": "czk",
      "price_max": null,
      "price_min": null,
    },
    "changed": "2017-11-02 14:21:18"
  }
}
]
}
}

```

---

#### Výpis 5: Ukázka výsledku vyhledávání

Uložení nastavení filtru probíhá následovně. Před odesláním formuláře parametrického filtru je konfigurace filtru ukládána do javascriptového objektu (Výpis 6). Po odeslání formuláře je konfigurace uložena z důvodu persistence do URL adresy. Tento způsob uložení je vhodný například v případě, že chceme vyplněný filtr a jeho výsledek sdílet nebo dočasně uložit odkaz na něj.

---

```

{
  "checkbox": {
    "46613": true,
    "46651": true
  },
  "checkboxList": {
    "46611": {
      "2274017": "2274017",
      "2274062": "2274062"
    }
  },
  "46609": {
    "2274018": "2274018",

```



```

        "2274045": "2274045",
        "2274050": "2274050"
    },
    },
    "range": {
        "46617": {
            "from": "2001",
            "to": "3500"
        },
        "46636": {
            "from": "150",
            "to": "20000"
        }
    },
    },
    "price": {
        "to": "444455",
        "from": "4444"
    }
}

```

---

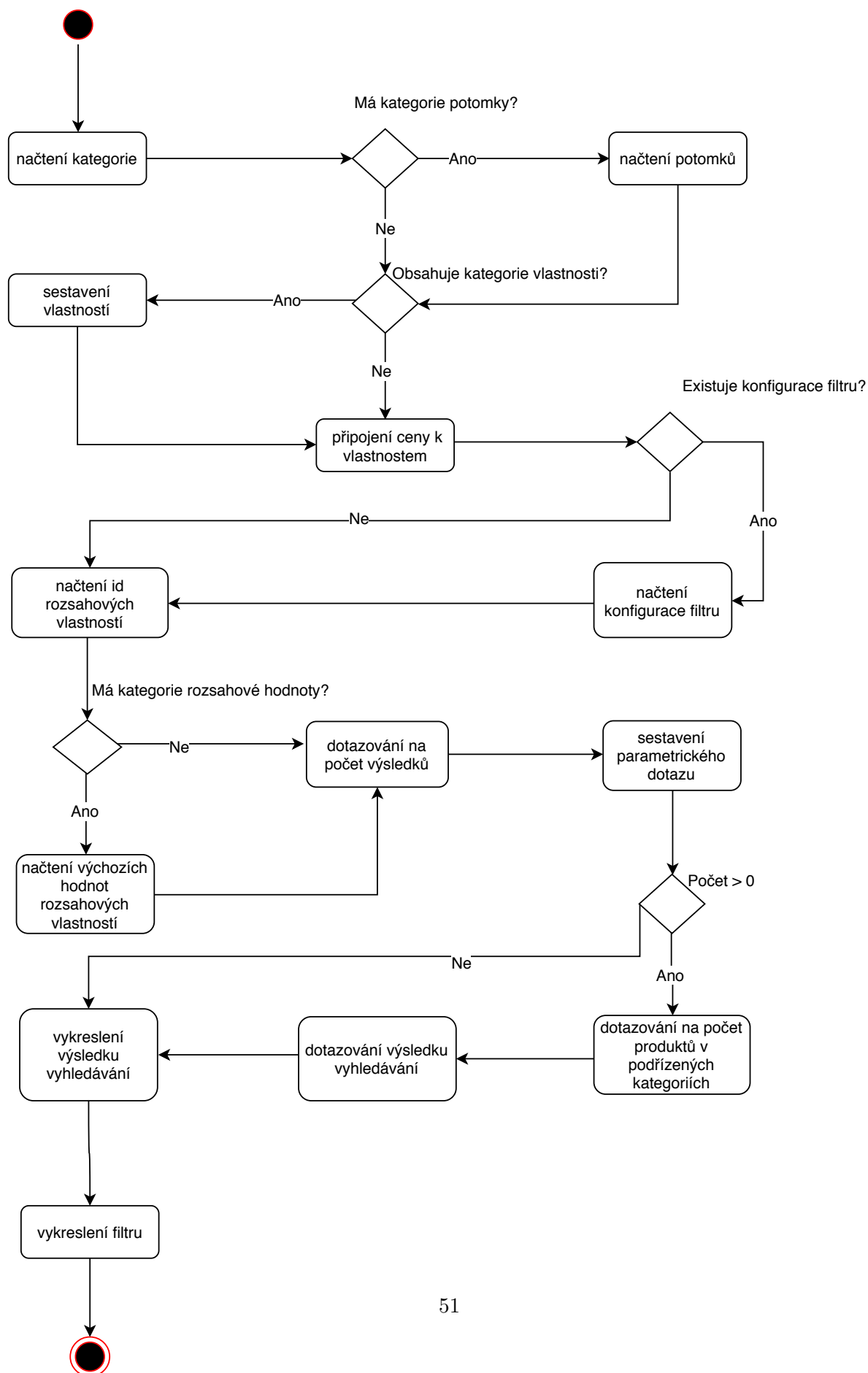
#### Výpis 6: Ukázka konfigurace filtru

Postup sestavení a odeslání filtru je následující:

1. Získání výchozích hodnot rozsahových vlastností filtru z existujících katalogových produktů
2. Vyplnění filtru z defaultních hodnot
3. Vyplnění filtru z existující konfigurace filtru definované uživatelem
4. Načtení existující konfigurace z URL parametru do javascriptového objektu reprezentující konfiguraci filtru
5. Vykreslení filtru
6. Vybrání nebo definování hodnoty parametru uživatelem
7. Uložení výběru do javascriptového objektu reprezentující konfiguraci filtru
8. Zaslání asynchronního dotazu na celkový počet katalogových produktů vyhovující aktuální konfiguraci a překreslení počtu
9. Odeslání vyhledávacího formuláře uživatelem přes tlačítko Zobrazit

10. Získání seznamu rozsahových vlastností nedefinovaných v konfiguraci z důvodu správného načtení výchozích hodnot
11. Sestavení dotazu na základě konfigurace filtru
12. Přechtení odpovědi Elasticu
13. Uložení konfigurace do URL
14. Překreslení stránky a zobrazení výsledku vyhledávání

Proces sestavení dotazu a vykreslení výsledku vyhledávání je také znázorněn diagramem 13.



Obrázek 13: Proces sestavení dotazu a vykreslení výsledku vyhledávání

#### 4.2.7 Cachování zpracovaných dat

Na základě nabytých informací z teoretické přípravy jsem byl schopen implementovat praktickou část cachování dat pomocí databáze Redis.

Pro tyto účely jsem implementoval třídu s názvem *RedisClient*, která slouží pro komunikaci s key-value databází Redis. Tato třída obaluje CRUD operace PHP rozšíření redis.

Při ukládání dat do Redisu je nutné zadat klíč a nastavit platnost záznamu uváděného obvykle v sekundách. Při definici klíčů jsem vycházel z konvencí pro tvorbu klíčů uvedených v kapitole 3.6. U vyhledávání jsem nastavoval platnost jednotlivých klíčů v závislosti na frekvenci změn dat. Pokud chceme data do Redisu uložit persistentně, použijeme hodnotu -1.

V rámci komponenty vyhledávání jsem do Redisu ukládal následující data:

1. Aktuální kategorie – Tento záznam obsahuje základní informace o aktuálně vyhledávané kategorii. Konkrétně se jedná o ID a přeložené popisky kategorie. Dále jsou zde uloženy všechny rodičovské kategorie aktuální kategorie. Klíčem je zde ve tvaru „category:id\_kategorie“.
2. Parametry a jejich vlastnosti – Sestavená kolekce dat pro aktuální kategorii. Kolekce je definována výše 4.2.4. Použitý klíč je ve tvaru „categoryProperties:id\_kategorie“.
3. Seznam rozsahových vlastností – Identifikátory rozsahových vlastností sloužící k získání výchozích hodnot do filtru. Klíč je ve formátu „categoryProperties:id\_kategorie:rangeIds“
4. Výsledek vyhledávání – Odpověď získaná z Elasticu na položený dotaz. Klíč je ve tvaru „catalogSearch:hash“. Hash je zde vytvořen z těla dotazu reprezentovaným textovým řetězcem za pomoci hashovacího algoritmu SHA-1.

### 4.3 Porovnání výkonu

V této části budu se zabývat ověřením správnosti hypotézy o cachování. Ta říká, že s cachováním dat dojde ke snížení požadavků na výpočetní výkon, kapacitu přenosu dat po síti a zrychlení poskytované služby.

Výkon bude porovnáván u základních operací parametrického vyhledávání. Pro porovnání výkonu je možné využít několik ukazatelů. V mém případě je ukazatelem výpočetní čas.

V následující tabulce se nachází naměřené časy:

Tabulka 3: Naměřené časy vykonávání při ověřování hypotézy cachování

Funkce	Čas bez cachování Redis [ms]	Čas s cache Redis [ms]
Sestavování datových podkladů	5620	184
Dotaz na počet výsledků	689	85
Dotaz na výsledek	5885	582
Celkový čas vykonávání	12194	851

Dle naměřených hodnot uvedených v tabulce 3 lze usoudit, že hypotéza cachování dat byla v tomto případě potvrzena. Využitím Redis se čas vykonávání sestavování datových podkladů snížil 30x, dotaz na počet výsledků 8x a dotaz na výsledek vyhledávání 10x. U Celkového času k vykonání všech sledovaných operací došlo ke 14-ti násobnému zrychlení.

Testování probíhalo na stoji, který se pro eliminaci ztrát síťového přenosu v době testování nacházel ve stejné síti jako databázové servery. Při testování bylo zjištěno, že nejvíce času zabírá proces sestavení datových podkladů a samotné vyhledávání. Podpůrné operace aplikačního serveru mají na celkový čas jen zanedbatelný vliv.

Vzhledem k relativně pomalým časům naměřených bez použití Redis je vhodné do budoucna počítat s předgenerováváním dat do cache systému a optimalizací dotazů prováděných v relační databázi.

## 5 Poptávka a nabídka

Následující kapitola bude zaměřena na problematiku poptávek a nabídek. V rámci analýzy aktuálního stavu budou definovány klíčové pojmy, existující poptávkové systémy katalogových produktů a párování produktů. V praktické části se budu věnovat jednotlivým navrženým a implementovaným druhům poptávek a návaznosti na parametrické vyhledávání. V rámci praktické části bude proveden také průzkum trhu pro určení, které typy poptávek jsou pro reálný provoz vhodné.

### 5.1 Analýza poptávek

V rámci analýzy poptávek je zapotřebí nejprve definovat pojem poptávka a nabídka. Následovat bude rozbor, co je to a jaký je princip poptávkového systému. Součástí jsou také existující řešení. Pro reálný provoz poptávkového systému budou popsány různé verze obchodního modelu. Poptávkový systém založený na poptávání katalogových produktů nemůže fungovat bez procesu zvaného párování produktů.

#### 5.1.1 Poptávka

Poptávka je definována jako vztah poptávaného množství a ceny, za kterou je nakupující toto množství ochoten zakoupit. Cena je v tomto vztahu určujícím faktorem. Princip poptávky spočívá v tom, že s rostoucím množstvím klesá cena za jeden kus. Existují tři druhy poptávek:

- Agregátní - Tato poptávka je dána celkovým objemem poptávané produkce a cenami, za které je zákazník tyto produkce odkoupit.
- Individuální - Jedná se o typ poptávky, kdy jeden kupující poptává jeden statek. Statkem se myslí cokoliv, co vyjadřuje nějakou hodnotu.
- Tržní - Sčítají se zde poptávky všech spotřebitelů po jednom výrobku nebo službě.

#### 5.1.2 Nabídka

Reakcí na poptávku je nabídka. Ta je definována jako vztah nabízeného množství statku a cenou, za kterou je výrobce/prodejce ochotný dodat poptávané množství statku na trh. Stejně jako poptávka je nabídka trojího typu:

- Agregátní - Nabídka je dána součtem objemů výroby všech výrobců a cenou, za kterou chtějí tyto výrobky prodat.
- Individuální - Při individuální nabídce nabízí jeden výrobce jeden konkrétní statek.
- Tržní - Všichni výrobci nabízejí jeden produkt.[18]

### 5.1.3 Poptávkový systém

Poptávkový systém je webový systém nebo mobilní aplikace, která umožňuje vytvářet internetovou komerci. Primární funkcionalitou poptávkového systému je vytvoření poptávky na produkt nebo službu a vytvoření nabídky reagující na konkrétní poptávku. Sekundární funkcí je například katalog firem. Pro fungování poptávkového systému je nezbytná široká základna zaregistrovaných dodavatelů, kteří budou reagovat na poptávky zákazníků. Pro uspokojení většiny poptávek je zapotřebí pokrýt všechny segmenty trhu.

Princip poptávkového systému spočívá v tom, že poptávající vytvoří poptávku na konkrétní produkt nebo službu. Tato poptávka je následně zaslána relevantním dodavatelům (zaregistrovaným firmám v systému). Následně tyto firmy zašlou nabídku klientovi. Pokud klient s nabídkou souhlasí, může dojít k realizaci obchodu.

Nejzásadnějším krokem z pohledu poptávkového systému je zajistit to, aby nabídka přišla pouze dodavatelům, kteří jsou schopni poskytnout svou službu nebo žádaný produkt v dostatečné kvalitě. Dodavatelé, kteří budou moci odpovídat na poptávku, bývají vybíráni na základě parametrů, které uvedly při registraci do systému. Zde může být například vybraný obor podnikání nebo běžný termín doručení. Dalším kritériem velmi často bývá kvalita získaného hodnocení z již realizovaných zakázek. Vytváření poptávek je většinou zdarma. Možnost reagovat na poptávku nabídkou už ale bývá zpoplatněno.

Z pohledu zpoplatnění nabídek poptávkového systému existuje hned několik modelů. Nejzákladnější jsou uvedeny zde:

- Nabídka zdarma - Proces poptávky a nabídky není zpoplatněn. Z toho důvodu je zapotřebí hledat finance na provozní náklady portálu, budou placeny z jiných zdrojů než z primární činnosti. Zdrojem může být například reklama při dostatečně velké návštěvnosti.
- Přístup k poptávkám - Za možnost reakce na poptávku a samotný přístup k nim firma zaplatí buďto paušální platbou na určité období, nebo fixní za jednu poptávku. Fixní cena se může měnit například dle kategorie poptávky.
- Výherní nabídka - Dodavatel zaplatí poptávkovému systému v době, kdy si jeho nabídku zákazník vybere. Zde je riziko pro dodavatele v tom, že si zákazník může obchod rozmyslet a nemusí reálně proběhnout. V tu chvíli dodavatel ztrácí vynaložené finance.
- Zrealizovaný obchod – Nejvýhodnější varianta pro dodavatele. Má přístup ke všem poptávkám zdarma a náklady na zákazníka mu vznikají až v době realizace obchodu. Jedná se o nejrizikovější variantu pro poptávkový systém, jelikož může dojít k jeho obcházení a nepřiznání uskutečněného obchodu. Zde je nutné dát pádný argument oběma stranám, aby k této situaci nedocházelo.

#### 5.1.4 Existující poptávkové systémy katalogových produktů

Analýza existujících systémů nebyla provedena, protože jsem v době vzniku této práce nenašel žádné konkurenční řešení, které by bylo adekvátně odpovídající vytvářenému. V potaz v České republice přicházely portály AAPoptávka.cz, epoptavka.cz, poptavej.cz, žádný z těchto portálů ale nepodporuje možnost poptávat konkrétní katalogový produkt.

#### 5.1.5 Párování produktů

Párování produktů je zpracování dat o produktech e-shopů. Data jsou poskytována v datovém souboru zvaném feed, který tyto informace přenáší. Nejběžnějším formátem feedu je XML. Jedná se o proces, při kterém dochází k mapování dat získaných od e-shopů na interní databázi katalogových produktů. Párování produktů je typická činnost produktových srovnávačů.

Párování může probíhat buď na úrovni nalezené přímé shody s katalogem produktů srovnávače, nebo na principu zařazení produktu do kategorie srovnávače. Typickými atributy, dle kterých párují data e-shopu české srovnávače, jsou EAN, název produktu, cena, název kategorie ve feedu.

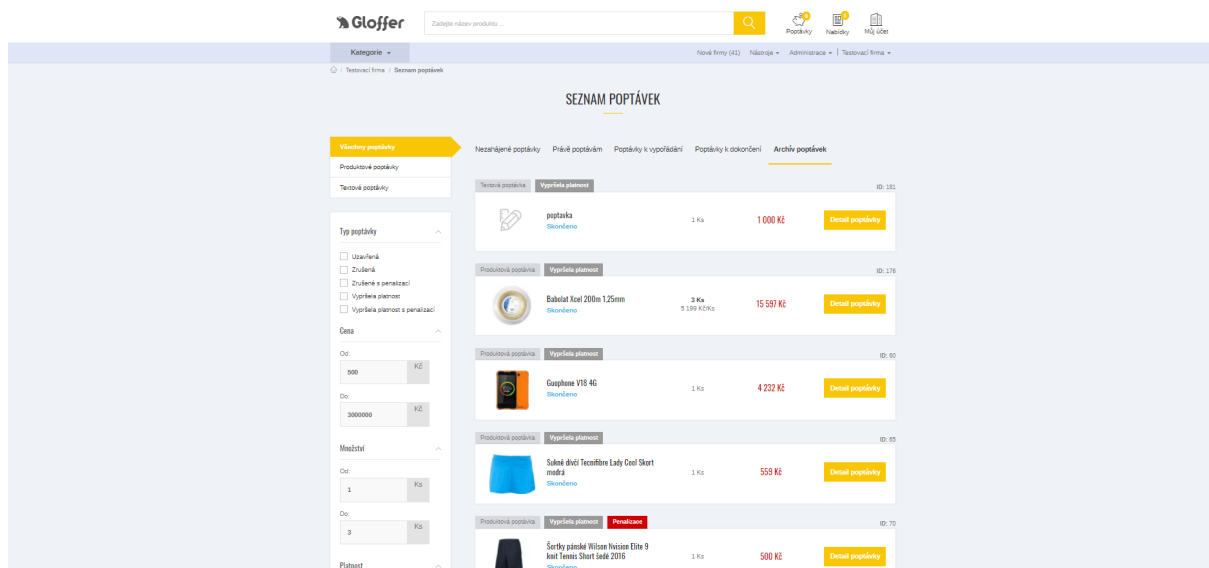
Obecný princip párování je následující. Nejprve aplikace zkusí spárovat data za pomoci EAN kódu na přesnou shodu. Pokud EAN není zadán, následuje párování na přesnou nebo částečnou shodu názvu produktu. Formát názvu produktu bývá definován konkrétním srovnávačem, do kterého jsou data nahrávána. Když se nepodaří nalézt shodu s katalogem produktů, dochází ke snaze zařadit produkt alespoň do správné kategorie, v opačném případě není produkt spárován.

Je důležité uvést, že párování dat je velice časově i výpočetně náročná činnost. Každý srovnávač využívá vlastní algoritmus pro párování dat a na něm leží výsledná kvalita a přesnost párování. Samozřejmostí je ale také kvalita vstupních dat, sebelepší algoritmus není schopen spárovat nekvalitní data.

## 5.2 Poptávky v systému Gloffer

Parametrické vyhledávání navržené a implementované v praktické části této práce má úzkou souvislost s poptávkovým systémem portálu (Obrázek 14). Existují různé způsoby, kterými můžeme poptávat. V rámci této práce vznikla poptávka na katalogový produkt a textová poptávka.

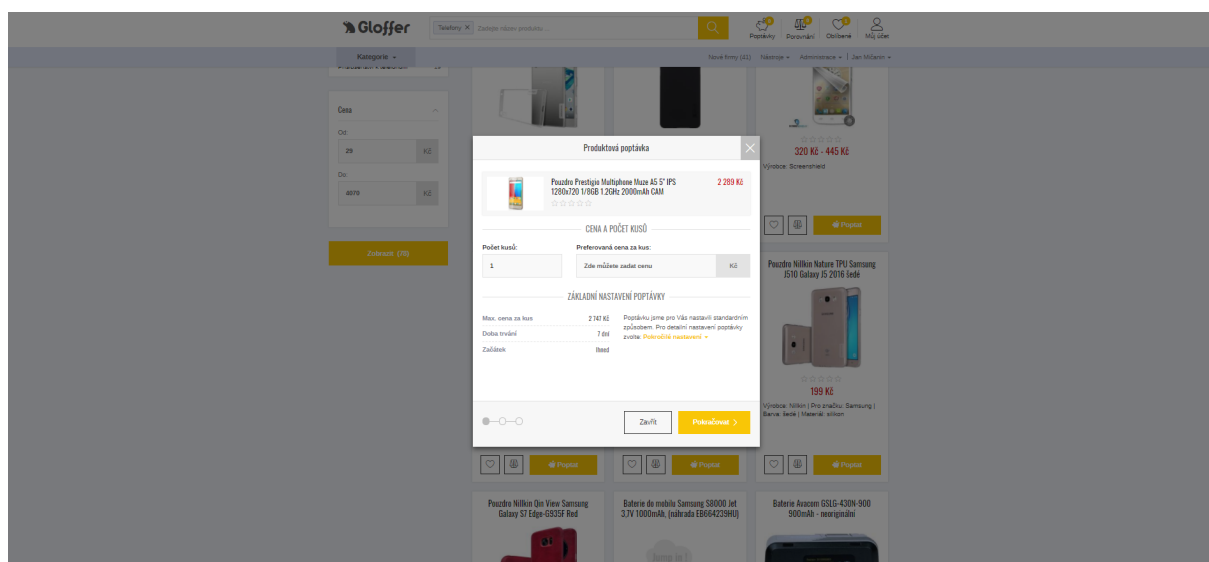




Obrázek 14: Poptávková sekce portálu Gloffer

### 5.2.1 Poptávka katalogového produktu

Prvním a nejpřesnějším způsobem je poptávka na katalogový produkt (Obrázek 15) Před samotným způsobem je ale zapotřebí žádaný produkt vyhledat. Při mé implementaci jsem využil cestu parametrického vyhledávání a zařazení produktů do kategorií. Jelikož parametrické vyhledávání poskytuje největší relevanci vyhledávaných produktů, jedná o hlavní způsob vytváření poptávek na katalogové produkty. Navrhl jsem a implementoval také komponentu, která umožňuje uživateli v systému jednoduše vyhledat produkty podobné. Podobnost je určována na základě shody klíčových vlastností dané kategorie.



Obrázek 15: Formulář pro vytvoření poptávky na katalogový produkt

### 5.2.2 Textová poptávka

Jedná se o druh poptávky, při které uživatel vypíše jednotlivé parametry produktu do textové zprávy. Tento poptávkový text je následně přečten odborníkem, jenž je kompetentní nabídnout relevantní produkt poptávajícímu. Textovou poptávku budou využívat především uživatelé, kteří nejsou schopni najít svůj ideální produkt za pomoci parametrického vyhledávání. Klíčem k úspěchu tohoto modelu je komunikace poptávajícího a nabízejícího před vytvořením samotné nabídky. Pro filtraci pouze relevantních poptávek na straně nabízejícího jsem vytvořil komponentu tzv. Sledovaný obsah. Účelem této komponenty je nastavení pravidel pro sledování obsahu a následné zobrazení pouze relevantních poptávek nabízejícímu. Sledovaný obsahem jsou myšleny kategorie produktů a jejich výrobci.

### 5.2.3 Další možnosti poptávání v systému

Aktuálně jsou v poptávkovém systému implementovány dva výše zmíněné způsoby poptávání. Samotný poptávkový systém je možné v budoucnu rozšířit o tyto konkrétní typy poptávek.

**5.2.3.1 Poptávka vlastního produktu** Vytvoření produktu na míru. Výhoda je v tom, že si zákazník sestaví produkt na přání. Nevýhoda tohoto řešení spočívá v situaci, kdy specificky nakonfigurovaný produkt neexistuje na trhu. V tom případě mu dodavatel může nabídnout pouze podobné/alternativní produkty.

**5.2.3.2 Poptávání feedového produktu** Tento způsob poptávání je možný u zboží, které systém nedokázal zařadit ke katalogovému produktu při procesu párování. Pro uživatele může být tato situace výhodná v případě, že požaduje od dodavatele nějaký benefit nebo slevu. Poptávka mu umožní jeho požadavky předat druhé straně. Nevýhodou tohoto řešení je to, že poptávku obdrží právě jeden dodavatel a odpadá tím možnost konkurenčního boje.

**5.2.3.3 Poptávka služby** V dnešní době se jedná prakticky o nejběžnější způsob poptávky. Tento způsob také nabízí klasické poptávkové systémy. Princip spočívá v textovém nebo parametrickém zadání poptávky na poptávanou službu. Většina poptávkových systémů funguje na principu, při kterém si zákazník vytvoří poptávku a následně zaměstnanci systému přeposílají požadavky zákazníků jednotlivým dodavatelům.

**5.2.3.4 Kombinovaná poptávka** Do jedné poptávky se zahrne více katalogových produktů nebo služeb. Výhoda spočívá v tom, že je možné požadovat množstevní slevu a je zajištěno, že veškeré zboží nebo služba jsou dodány od jednoho dodavatele. To zjednodušuje případné reklamace.

## 6 Správa uživatelských požadavků a komunikace se zákazníky

Při vývoji softwaru je důležité zajistit komunikační bod se zákazníky. V případě informačních systémů je tato komunikace zajištěna přes systém pro správu uživatelských požadavků. V této kapitole bude provedena analýza aktuálního stavu. Na základě této analýzy bude vytvořen návrh a implementace vlastní správy požadavků.

### 6.1 Analýza komunikace s uživateli

V rámci analýzy aktuálního stavu se budu věnovat definici základních pojmů jakými jsou požadavek a tiketovací systém. Pro porozumění dané problematiky bude sloužit také výčet zásad komunikace se zákazníky. Následovat bude popis a srovnání vybraných existujících řešení na trhu, konkrétně se jedná o systémy Freshdesk, Zendesk a Jira.

#### 6.1.1 Tiket

Tiket, jiným slovem požadavek, slouží v tiketovacím systému pro komunikaci zákazníka s technickou podporou provozovatele služby. V rámci tiketu se mohou řešit například problémy s používáním aplikace, hlášení chyb, náměty na zlepšení nebo jen prostý dotaz uživatele. Každý tiket v systému má obvykle přidělené jedinečné identifikační číslo, které se po dobu komunikace nemění. Je nezbytné, aby tiket vždy obsahoval název, identifikační údaje odesílatele, text zprávy a kategorii, do které má být tiket zařazen.

### 6.2 Tiketovací systém

Tiketovací systém je souhrnný název pro aplikaci, jejíž hlavní náplní je vytváření a vybavování tiketů. Systém má většinou dva typy uživatelů. Prvním je zákazník, který vytváří samotné požadavky. Na druhé straně se nachází osoby odpovědné za jejich vyřizování. Může se jednat o běžné zaměstnance helpdesku, kteří vyřizují požadavky uživatelského charakteru, jako je registrace, vytvoření platby atp., nebo přímo o administrátora systému. Administrátor vyřizuje požadavky technického charakteru, například nahlášení chyb. U menších systémů se role zaměstnance helpdesku a administrátora spojuje do jednoho.

#### 6.2.1 Zásady komunikace se zákazníky

Správná komunikace se zákazníkem je jedním z hlavních opěrných bodů úspěšného podnikání. Když prodejce v zákazníkovi nabudí pozitivní pocit, že je s ním zacházeno slušně, zvyšuje se šance, že zákazník nakoupí a bude se opětovně vracet. Existuje několik hlavních zásad komunikace se zákazníkem:

- Rychlost odpovědi - První zásadou je zákazníkům odpovídat co nejrychleji a systematicky. Pokud budou zákazníci dlouho čekat, reputaci to neprospěje.

- Slušný a otevřený způsob komunikace - Druhou zásadou je jednat se zákazníkem vždy slušně a nevyvolávat zbytečné konfrontace. Na všechny dotazy je zapotřebí odpovídat pravdivě. Důležité je také odpovídat i na nepříjemné dotazy a diskutovat s nespokojenými zákazníky.
- Konzistentnost stylu komunikace - V tomto bodu je důležité si zadefinovat, jak budeme se zákazníkem komunikovat, zdali mu budeme vykat či oslovovat křestním jménem. Je nezbytností zavést jednotnou patičku odpovědí pro všechny komunikační kanály.
- Dobrá nálada - Posledním doporučením je zachovat vždy dobrou náladu a nepřenášet na zákazníky aktuální stav prodávajícího. V telefonní komunikaci je zapotřebí vystupovat vždy příjemně, totéž platí pro písemnou.[23]

## 6.2.2 Existující tiketovací systémy

V dnešní době mají vývojáři informačních systémů dva možné způsoby, jak vyřešit komunikaci s jeho uživateli. První volbou je využít stávajících existujících řešení, které nabízejí třetí strany. Druhou je vytvoření vlastního tiketovacího systému na míru vlastním potřebám. V následujících odstavcích popíšu možná existující řešení, konkrétně Freshdesk, Zendesk, JIRA a jejich porovnání. Druhé volbě se budu věnovat v praktické části této práce.

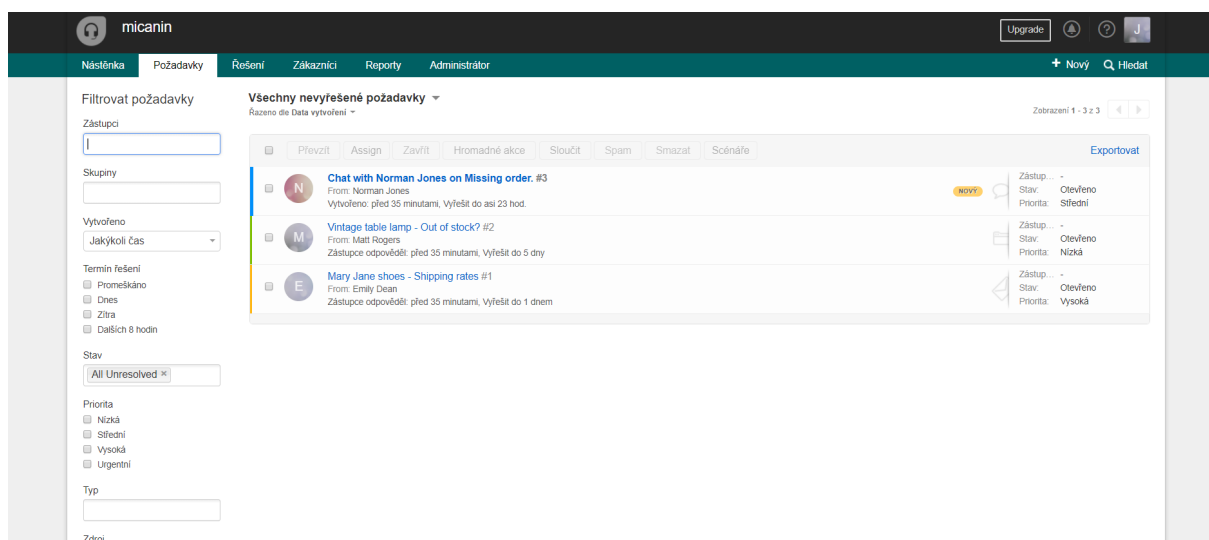
**6.2.2.1 Freshdesk** Jedná se o první testovaný tiketovací systém provozovaný firmou Freshworks Inc., který byl založen roku 2010. V roce 2011 vyhrál projekt soutěž Microsoft BizSpark Startup challenge. Přelomový bod byl v roce 2015, kdy Freshdesk získal od investorů 50 milionů dolarů a následně Freshdesk oslavil milník 50 000 zákazníků. V dnešní době se jedná o vůbec nej používanější tiketovací systém vůbec.

Z hlediska funkcionalit Freshdesk nabízí klasickou škálu funkcí jakými jsou například dashboard (Obrázek 16) se základními statistikami, seznam tiketů, filtrování, řazení, změnu stavu tiketů, přidělování odpovědných osob, reporty, seznamy zákazníků. Na detailu tiketu je vyobrazen strom komunikace (Obrázek 17).

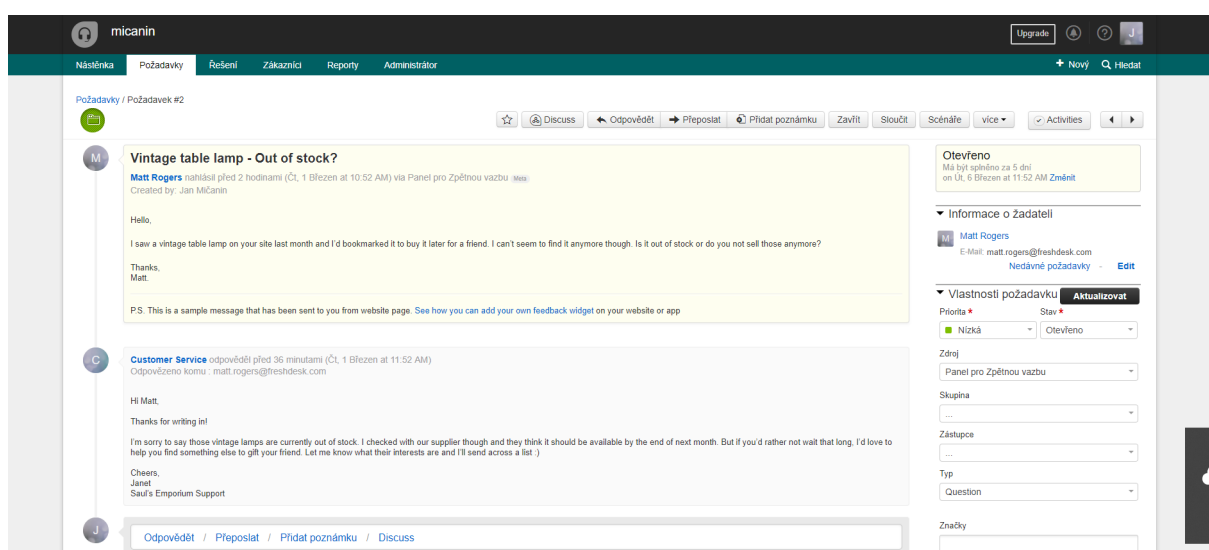
Kromě základních služeb má Freshdesk také několik dalších užitečných funkcionalit:

- Mobilní aplikace - Umožňuje vyřizování tiketů na cestách.
- Feedback widget - Formulář, který si může zákazník místit na svůj web a umožnit tak svým klientům zanechat zpětnou vazbu.
- Live chat - Funkce nabízející komunikaci zákazníkovi s klienty v reálném čase a umožňující řešení problémů ihned.
- Facebook and Twitter - Komponenta pro správu konverzací na sociálních sítích.

Z pohledu lokalizace pro Českou republiku je systém částečně přeložen do češtiny, zbytek popisků zůstává ve výchozím jazyce.[19]

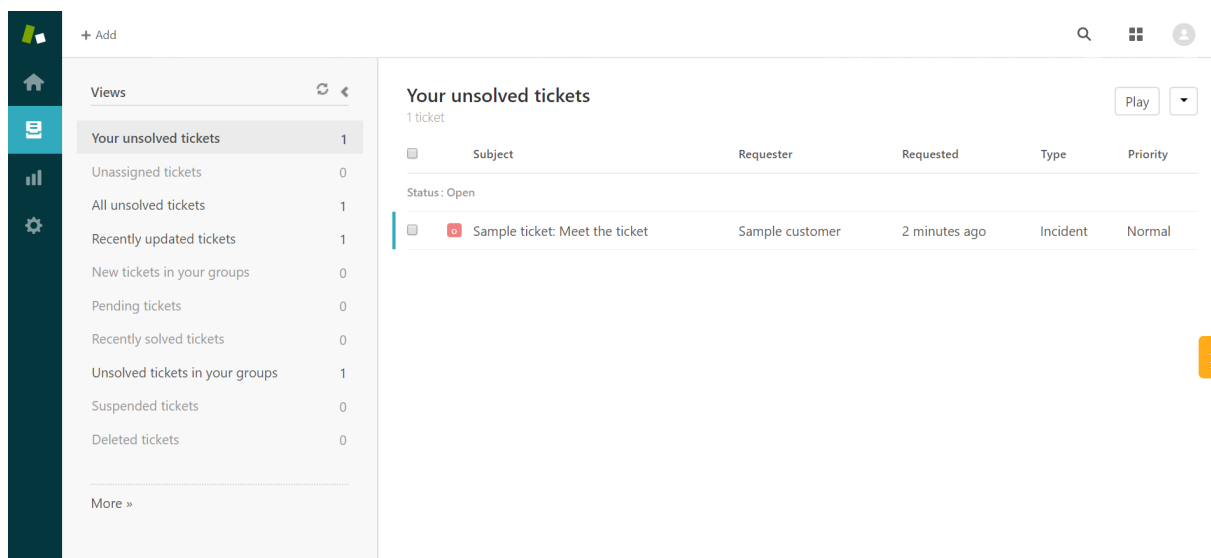


Obrázek 16: Dashboard systému Freshdesk

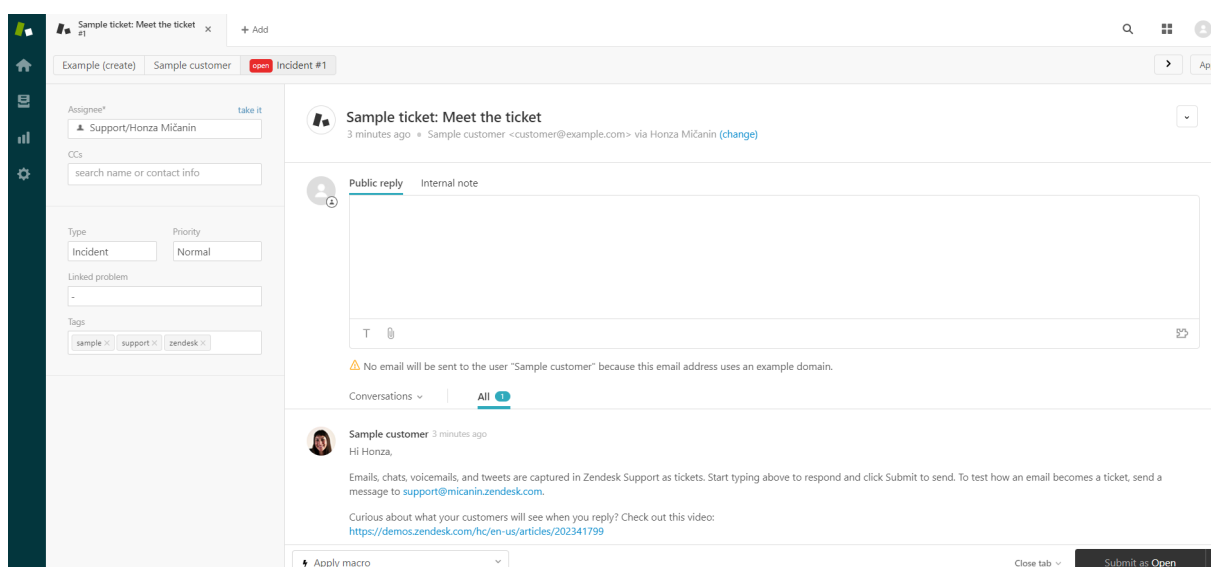


Obrázek 17: Detail tiketu systému Freshdesk

**6.2.2.2 Zendesk** Zendesk byl druhý z testovaných systémů. Vznikl v roce 2007 v Dánsku jako myšlenka tří kamarádů. V dnešní době má Zendesk ústředí v USA a zaměstnává 2 000 lidí. Používá ho pro správu komunikace se zákazníky přes 20 000 platících společností po celém světě a dosahuje tržeb přes 430 milionů dolarů. Ve svém jádru je skoro stejný jako výše popsany Freshdesk. Pro vizuální porovnání s Freshdesk uvádím snímek dashboardu (Obrázek 18) a detailu tiketu (Obrázek 19). Pro usnadnění práce se systémem poskytuje video nápovědy. [20]



Obrázek 18: Dashboard systému Zendesk



Obrázek 19: Detail tiketu systému Zendesk

**6.2.2.3 JIRA** Třetím a posledním testovaným systémem byl produkt s názvem Jira vyvíjený společností Atlassian. První verze vznikla už v roce 2004. Jira je nástroj pro řízení projektů a evidenci chyb. Správa požadavků uživatelů je jen jedna z mnoha funkcionalit. Jira dále nabízí například podporu workflow projektového vývoje, projektové statistiky, sdílenou komunikaci členů týmu, správu úkolů firemních zaměstnanců a spolupráci v reálném čase. Samozřejmostí je fulltextové vyhledávání a kvalitní filtrování obsahu systému.[21]

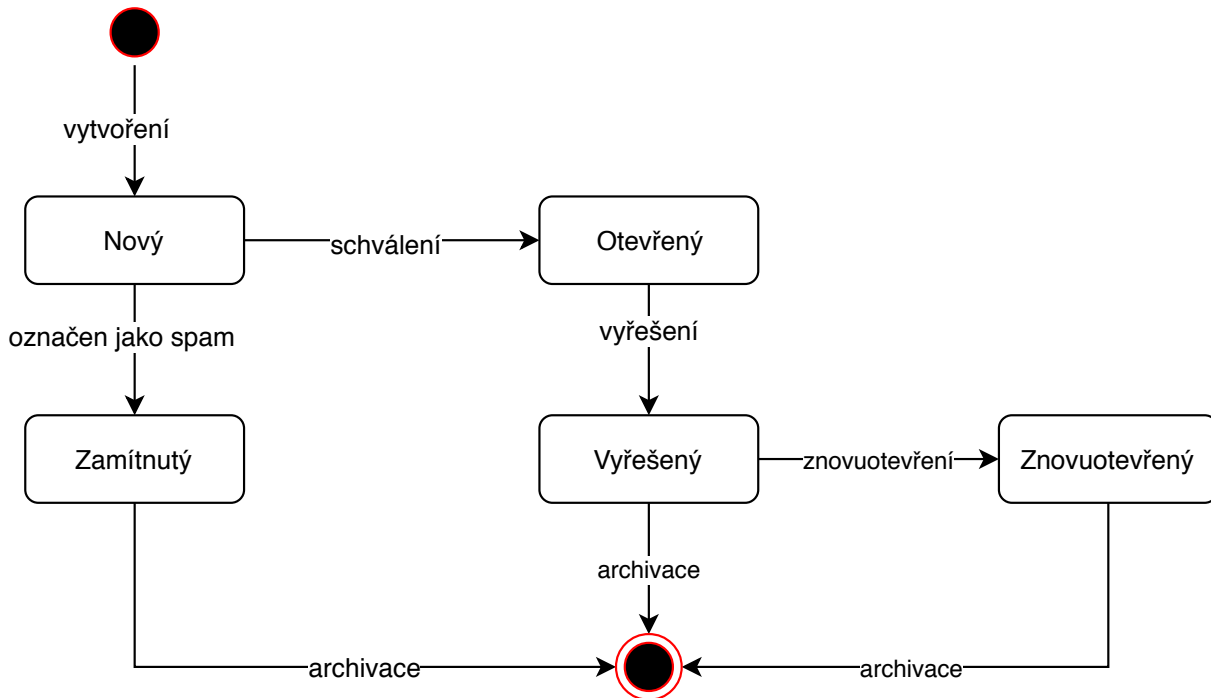
### 6.2.3 Srovnání existujících řešení

Stejně jako Freshdesk je možné si Zendesk vyzkoušet na 30 dní zdarma. Naproti tomu systém JIRA v Cloud řešení je možné využívat zdarma pouze 7 dní. Freshdesk i Zendesk podporují komunikaci přes sociální sítě Facebook a Twitter, Jira nikoliv. Zendesk oproti Freshdesku nabízí detailnější reporty a statistiky s grafy. Výhoda Freshdesku oproti konkurentům je v podpoře češtiny. Při přímém srovnání seznamu tiketů a detailu tiketu Freshdesku i Zendesku vychází subjektivně přehledněji řešení od Freshdesku. Výhodou JIRA oproti konkurentům je možnost nainstalovat si software na vlastní server.

Obecně je možné říci, že v případě požadavku klasického helpdesku je lepší využít služeb Freshdesk a Zendesk. Pokud se zákazník rozhodne pro projektové řízení využít systém JIRA, bude pro něj výhodnější využít tento systém i pro správu uživatelských požadavků. Jedna z nevýhod JIRA je v tom, že v případě narůstajícího počtu správců helpdesku také masivně narůstá paušální platba za tuto službu (cena se pohybuje v řádu tisíců dolarů).

### 6.2.4 Specifikace požadavků tiketovacího systému

V systému budou existovat tři typy uživatelů, kteří mohou vytvořit požadavek v systému. Každá role v systému má jasně definované způsoby, jakým mohou tiket vytvořit. Případy užití jednotlivých rolí jsou znázorněny na obrázku 21. Každý požadavek má v dané fázi nastaven stav daný diagramem 20.

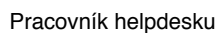


Obrázek 20: Stavový diagram požadavku

Role definované v systému jsou následovné:

- Nepřihlášený uživatel - Tato role má dva způsoby, jak servisní požadavek vytvořit. Prvním způsobem je zaslání emailu na systémem definovanou e-mailovou schránku. Druhou možností; je využít formulář pro vytvoření požadavku přímo v systému. Nepřihlášenému uživateli je zaslána odpověď na e-mail, který specifikoval ve formuláři, nebo ze kterého byl zaslán původní požadavek.
- Registrovaný uživatel - Uživatel v této roli má stejné možnosti pro vytvoření tiketu jako nepřihlášený uživatel. Navíc oproti této roli má možnost vytvořit tiket přes svou správu požadavků.
- Pracovník helpdesku - Tato role nemůže vytvářet servisní požadavky stejnými způsoby jako výše uvedené role. Jediným způsobem, jak požadavek může vzniknout je ten, že je automaticky vytvořen po zaslání emailů přes administrátorskou část helpdesku. Požadavek je vytvořen z důvodu, aby mohl adresát emailu odpovědět na email zaslaný z helpdesku a nedošlo z jeho strany k založení nového tiketu.





65

## 6.3 Návrh a implementace modulu uživatelských požadavků

Při návrhu jsem využil zkušenosti nabyté analýzou existujících řešení na trhu.

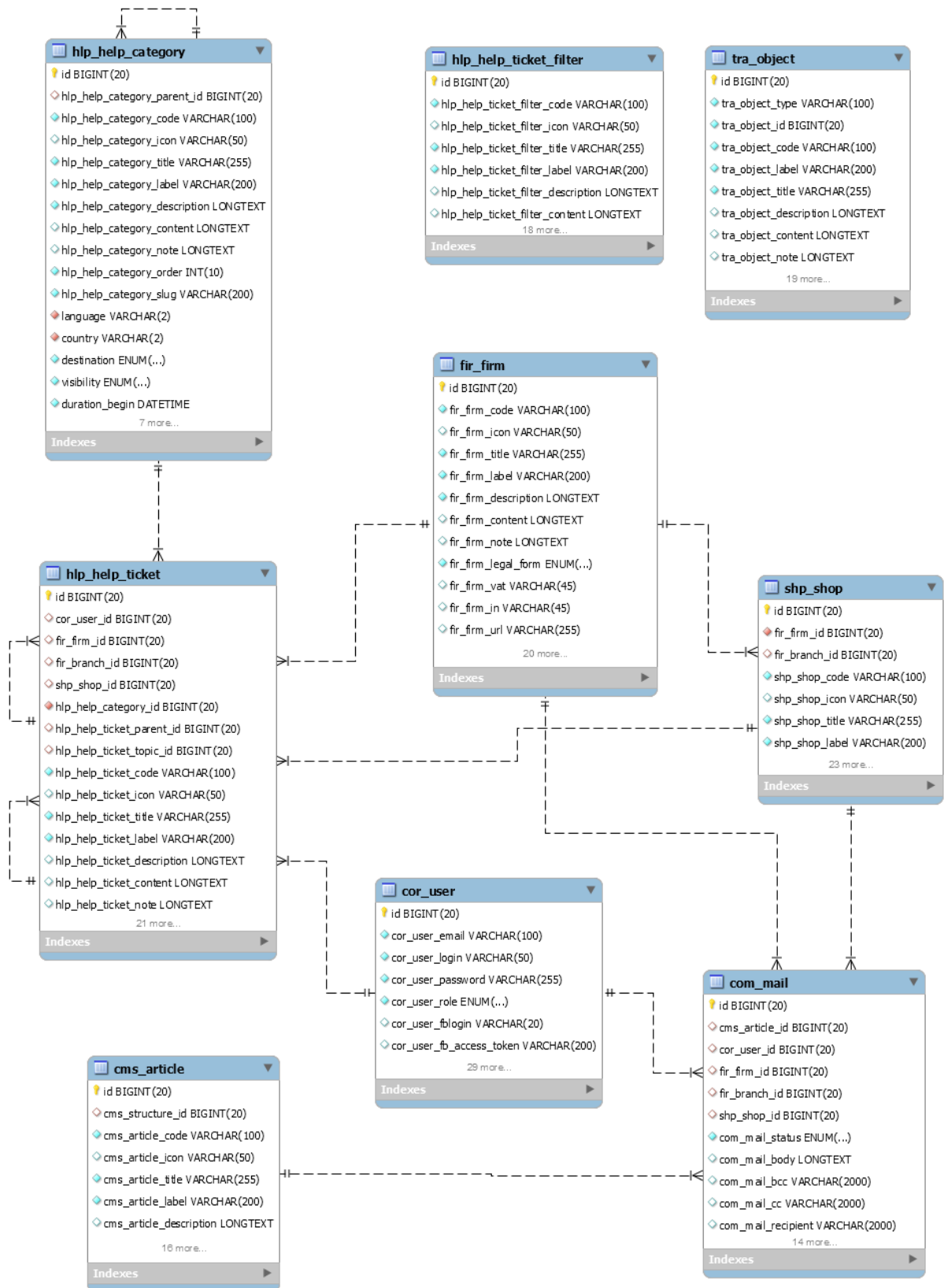
V této části práce jsem navrhoval a implementoval veškeré základní funkcionality helpdesku. Konkrétně jsem vytvořil část zabývající se základními operacemi s požadavky v systému jakými jsou vytvoření, odstranění, uzavření, označení požadavku jako spam, označení požadavku nebo odpovědi na něj jako přečtené a vytvoření odpovědi na tento požadavek. Součástí helpdesku je také komponenta umožňující odesílání, stahování a zařazování e-mailové komunikace do vnitřního systému tiketů.

### 6.3.1 Návrh databázové struktury

Návrh databázové struktury obsahuje entity reprezentující uložené požadavky, blacklist e-mailových adres a domén a entity týkající se e-mailové komunikace. Zahrnuty jsou ve struktuře také role uživatelů systému. Vazby mezi entitami jsou vyobrazeny na obrázku 22.

Seznam použitých tabulek:

- hlp\_help\_ticket – Požadavky a odpovědi
- hlp\_help\_category – Kategorie požadavku
- hlp\_help\_ticket\_filter – Blacklist emailových adres a domén požadavků
- com\_mail – Logování emailů
- cms\_article - Šablona (článek) pro email
- tra\_object – Tabulka s překlady všech objektů včetně článků
- cor\_user – Uživatel systému
- fir\_firm – Firma v systému
- shp\_shop – Obchod v systému



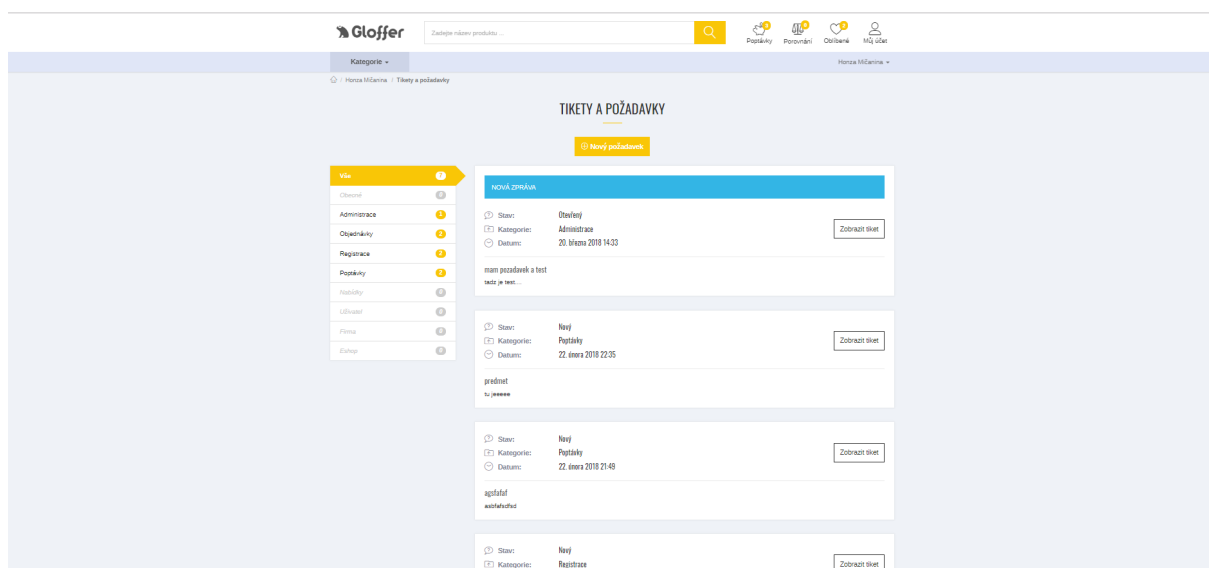
Obrázek 22: ER diagram tiketovacího systému

### 6.3.2 Uživatelská část

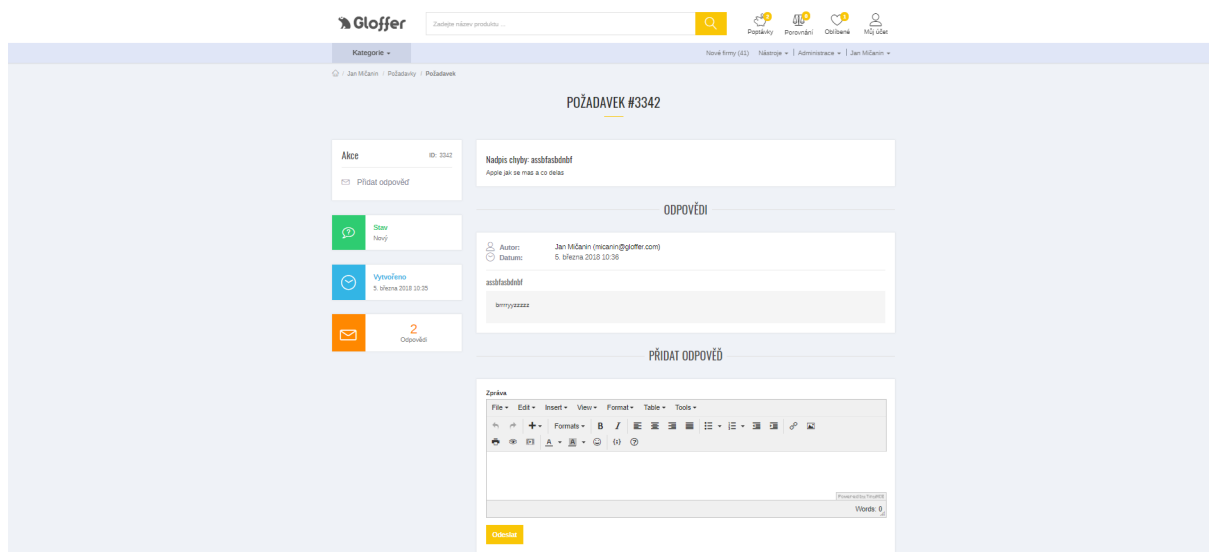
Uživatelská část helpdesku je určena pouze zaregistrovaným uživatelům a je rozdělena na několik základních částí:

**6.3.2.1 Přehled tiketů uživatele** V této sekci se nachází přehled vlastních požadavků uživatele (Obrázek 23). Konkrétně je zde uveden stav tiketu, kategorie, datum vytvoření, nadpis a obsah tiketu. Je se zde také odkaz na detailní informace o daném požadavku. Součástí je jednoduchý filtr požadavků dle aktuálního stavu. Na stránce s přehledem se také nachází odkaz s formulářem na vytvoření nového tiketu.

**6.3.2.2 Detail tiketu uživatele** Na detailu (Obrázek 24) se nachází všechny detailní informace daného tiketu. Součástí je také historie komunikace daného požadavku a formulář pro odpověď.



Obrázek 23: Přehled tiketů uživatele



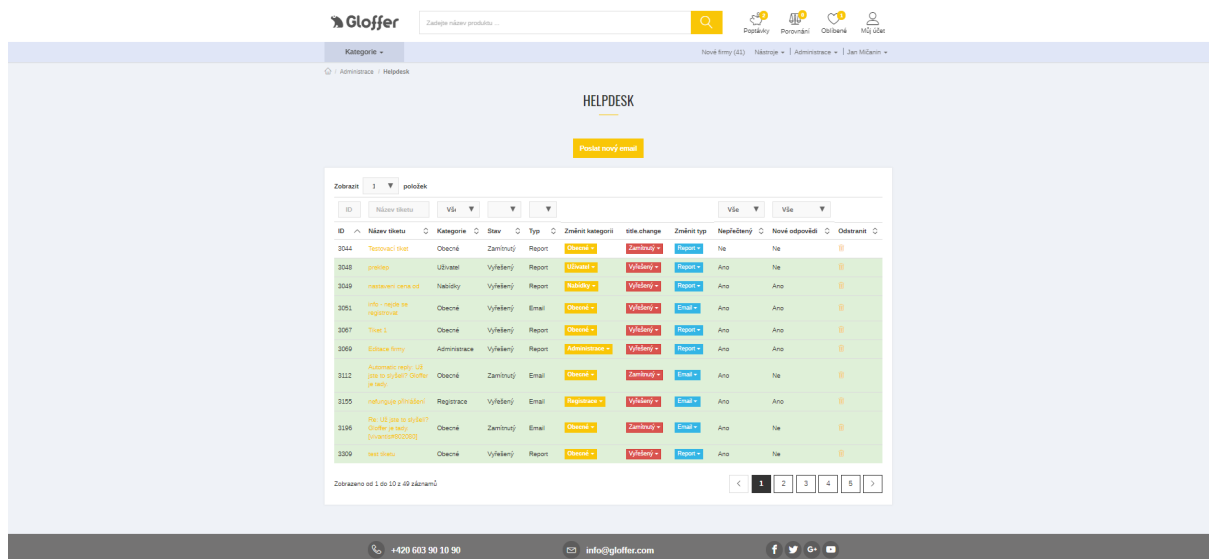
Obrázek 24: Detail tiketu uživatele

**6.3.2.3 Vytvoření nového tiketu** Nový tiket v systému vytváří uživatel prostřednictvím formuláře k tomu určenému. Další možností je zaslání emailu na specifikované adresy portálu Gloffer.

### 6.3.3 Administrátorská část

Správcovská část tiketovacího systému je podobně jako uživatelská rozdělena na několik sekcí.

**6.3.3.1 Přehled tiketů v administraci** Jako dashboard tiket systému je využit datový grid (Obrázek 25), který umožňuje filtrování, řazení a provádění základních operací s tikety. Konkrétně se jedná o změna stavu, zdroj, kategorie a smazání tiketu.



Obrázek 25: Přehled tiketů v administrátorské části tiketovacího systému

**6.3.3.2 Detail tiketu v administraci** Detail tiketu oproti uživatelskému obsahuje ještě rozšířené akce a informace o požadavku. Z akcí se zde nachází funkce pro změnu stavu tiketu, označení tiketu jako spam a potvrzení přečtení požadavku. Detail dále obsahuje URL adresu, ze které byl tiket odeslán (důležité v případě hlášení chyb), a informace o prohlížeči uživatele.

**6.3.3.3 Vytvoření nového tiketu** Správce helpdesku nemá možnost vytvořit tiket stejným způsobem jako přihlášený uživatel. Má pouze možnost zaslat email zákazníkovi, a tak dojde k vytvoření tiketu z důvodu napárování případné odpovědi na správné vlákno požadavku. Pro odeslání slouží formulář na vytvoření nového e-mailu (Obrázek 26). Při vytváření může pracovník zvolit možnost využít existující šablony v systému.

Veškeré tikety v systému jsou z implementačního hlediska vytvářeny stejným způsobem. Mění se pouze atribut, který definuje, zdali byl požadavek vytvořen správcem nebo běžným uživatelem. Metoda pro vytvoření tiketu se nazývá *createNewTicket* a nachází se ve třídě s názvem *HelpTicketManager*.

Položky označené \* jsou povinné.

Šablona

(57) Za tuhle cenu to stojí jistě za to!

Odesílatel \*

Email adresáta \*

Jméno adresáta

Kopie

Vzor: "vzor@gloffer.com; vzor2@gloffer.com; vzor3@gloffer.com"

Skrýtá kopie

Vzor: "vzor@gloffer.com; vzor2@gloffer.com; vzor3@gloffer.com"

Předmět \*

Zpráva \*

File Edit Insert View Format Table Tools

Formats B I

Konkurenční portály staví své business modely na tom, že z Vás tahají peníze, i když se obchod neuskuteční. [[icon: money; label: Různá řešení]] Gloffer nabízí i jiné řešení, které určitě zvýší Vaši spokojenost i výdělek. [[icon: user-tie; label: Žádné provize]] U nás nejsou žádné provize, neplatíte ani za zprostředkování [[icon: addressbook; label: Vše je veřejné]] Všechny kontakty jsou veřejné, nemáme před Vámi žádné tajnosti. [[icon: mobile3; label: Nepřetržitá komunikace]] Komunikujete s kolika zákazníky chcete, kdy chcete a jak chcete. [[icon: moneybag; label: Férový systém]] Gloffer je férový systém, který nemusíte obcházet, abyste ušetřili. [[icon: creditcard; label: Neplatíte za úspěch]] Je jedno, jak velký obchod uděláte, u nás provizi z prodeje prostě neplatíte. [[icon: cursor; label: Neplatíte za proklik]] Neplatíte ani za proklik na Vaše stránky, proč platit za to co je přirozené. [[icon: database2; label: Paušální ceny]] Nabízíme paušální ceny, které rozšíří Vaše možnosti a sníží náklady na získání jednoho zákazníka. [[icon: credit-card; label: Dva měsíce zdarma]] A navíc pokud zaplatíte na rok dopředu, získáte 2 měsíce zdarma. [[icon: matric; label: end]]

Powered by Thrift

Words: 173

Odeslat

Obrázek 26: Vytvoření nového emailu prostřednictvím tiketovacího systému

### 6.3.4 Odesílání emailů

V portálu Gloffer jsem vytvořil komponentu, která slouží pro odesílání e-mailů. Komponenta využívá pro zaslání a připojení s SMTP serveru nativní knihovnu frameworku *NetteMail*.

Vytvořit a zaslat email je možné dvěma způsoby. Prvním je zaslání přes formulář v helpdesku administrátorem, kdy je celé tělo definováno položkami daného formuláře. Druhou, více používanou, metodou je využití šablony e-mailu (šablona je v systému uložena jako článek), ve které jsou nahrazené proměnné jako například oslovení uživatele, jméno firmy, aktivací odkaz atd. Proměnná je definována ve formátu „%proměnná%“. Příklad článku je uveden ve výpisu 7.

Skrze tuto komponentu je kromě helpdesk e-mailů také zasílána všechna emailová komunikace poptávkového systému.

V práci jsem neimplementoval vlastní SMTP server, ale využil jsem server SMTP portálu svethostingu.cz. Při testování zasílání e-mailů jsem narazil na situaci, kdy došlo k překročení limitů zaslaných zpráv za den. Tento problém jsem vyřešil kontaktováním technické podpory hostingu a navýšení limitů.

Veškeré zaslané e-maily jsou logovány do relační databáze. Ukládají se zde informace o odesílateli, příjemci, použité šablony, předmětu a tělu e-mailu. V případě, že dojde k chybě při odesílání e-mailu ze strany SMTP serveru, je zde také uloženo chybové hlášení.

Řídící třída komponenty je umístěna v souboru *EmailManager*. Pro odeslání e-mailu slouží metoda *sendEmail*.

---

`%login%, welcome between the successful people!`

`Your request about registration was confirmed.`

`New %object_type% (%object_id%): %object_title%`

`Is now available in Gloffer system and you can start to use it.`

`Implementation team of Gloffer.com`

---

Výpis 7: Šablona těla e-mailu

### 6.3.5 Propojení e-mailové komunikace s tiketovacím systémem

Pro účely vytvoření požadavku nebo spárování odpovědi na něj prostřednictvím emailové komunikace jsem implementoval komponentu, která zajišťuje stahování dat z e-mailové schránky. Tato slouží pro sjednocení a zjednodušení komunikace se zákazníky portálu Gloffer.

Připojení je zajištěno protokolem IMAP se zabezpečením SSL. Pro samotnou komunikaci se serverem jsem použil vestavěnou PHP knihovnou *PhpImap*.

Na začátku procesu probíhá připojení k poštovní schránce. Následně jsou staženy ze schránky všechny nepřečtené e-maily. Poté je načten seznam blokováných domén a e-mailových adres z interní databáze systému. Následně jsou všechny emaily procházeny v cyklu. Pokud je nalezena shoda s blacklistem, je e-mail přesunut v e-mailové schránce do složky *tiketspam*. V opačném případě začne proces párování, při kterém se vyhledává identifikátor existujícího požadavku ve specifickém formátu „*#HT:id\_pozadavku*“. Ten je vždy lokalizován buď v předmětu e-mailu, nebo v patičce. Pokud se nepodaří e-mail spárovat s existujícím požadavkem, dojde k vytvoření nového. Dle názvu e-mailové adresy, na kterou byl e-mail zaslán, dojde k jeho zařazení do skupiny požadavků. Do stejné složky je přesunut v rámci e-mailové schránky.

Všechny stažené e-maily jsou označeny ve schránce za přečtené.

Celý proces stahování e-mailové komunikace a párování na existující tikety v systému je aktuálně spouštěn při každém zobrazení tiketu v administrativní části helpdesku. V budoucnu



bude tento proces spouštěn v rámci Cronu v pravidelném intervalu. Při stahování e-mailů dochází před uložením do databáze k odstranění nebezpečných HTML značek a escapování všech textů.

Hlavní funkcionality procesu se nachází ve třídě *pushTicketsFromMailboxToDatabase*, která se nachází v souboru *MailboxManager*.

## 7 Online platby a objednávání služeb

Kapitole online platby a objednávání služeb je posledním hlavním bodem diplomové práce. V rámci analýzy budou popsány možnosti online plateb v České republice. Z praktického hlediska bude proveden návrh a implementace procesu objednávání a správy objednávek a služeb. Pro párování zaplacených objednávek bankovním převodem bude implementována komunikace s Fio bankou. Pro platby kartou online bude využita služba GoPay.

### 7.1 Analýza aktuálního stavu

V dnešní světě internetového nakupování, kdy jsou zákazníci zvyklí dostávat zboží a služby v co nejkratším čase, je velice důležité využívat online platby.

V zásadě v online platbách jsou uživatelé rozděleni na dvě skupiny. První skupinou je nakupující, či zákazník, který si přeje zaplatit za službu nebo zboží obchodníkovi. Obchodníkem je ten uživatel, který přijme platbu od nakupujícího.

Nejdůležitějším bodem online plateb je zabezpečení transakce, tj. bezpečný přesun financí z účtu nakupujícího na účet obchodníka. Z důvodu redukce nákladů obchodníků v oblasti vývoje a implementace platebních metod ve vlastním vznikla služba zvaná platební brána. Ukazatelem její kvality může být plnění PCI DSS standardu.

Analýza aktuálního stavu je rozdělena na dvě části. V první se nachází popis možností online plateb. V další sekci bude popsán obchodní model portálu Gloffer.

#### 7.1.1 Možnosti online plateb

V následující kapitole bude tato služba popsána a budou zde uvedeny také další možnosti placení online v České republice.

**7.1.1.1 Platební brána** Platební brána je aplikace, jejíž úkolem je realizace online platby. Slouží jako agregátor platebních metod. V České republice existuje několik platebních bran, konkrétně se jedná o GoPay, PayU, ComGate, ThePay, ČSOB, Fio, Česká spořitelna, Komerční banka, UniCredit Bank, Raiffeisenbank, GP webpay, Pays a W-platba.

Při rozhodování obchodníka, kterou platební bránu zvolit, hraje v Čechách velkou roli zavedení EET při platbě kartou. Komunikaci s Finanční správou a vystavování EET účtenek za obchodníka řeší jen některé platební brány, konkrétně se jedná o GoPay, ComGate, ThePay, ČSOB, FIO, Komerční bank a Pays, u ostatních musí komunikaci s Finanční správou řešit obchodník sám. Dalším důležitým faktorem při výběru platební brány je splnění tzv. PCI DSS certifikace. V neposlední řadě hraje také důležitou roli při rozhodování, kterou platební bránu zvolit, její platební model a spektrum poskytovaných platebních metod, mezi které patří například platba kartou online, bankovní převod, online bankovní tlačítko či mladé platební metody,

jakými je elektronická peněženka, digitální mobilní peněženka, kupónová platba nebo mobilní platba.[24]

**7.1.1.2 PCI DSS standard** Bezpečnostní standard představuje mezinárodní pravidla, která definují podmínky uložení a práci s daty majitelů platebních karet. Dodržování pravidel je vyžadováno kartovými asociacemi a společnostmi po organizacích, které zpracovávají a uchovávají data platebních karet a jejich transakcí. Ty mají následně nárok na vystavení certifikátu. Při kontrole, zdali organizace plní mezinárodní pravidla, je kladen největší důraz na bezpečnost uložených dat jak fyzickou, tak kybernetickou.

Například z důvodu plnění požadavků na fyzické zabezpečení převedla společnost GoPay svá řešení do cloudového datového centra Amazon Web Services. U bankovních institucí, které chtějí poskytovat platby za pomocí platebních karet, je tento soubor pravidel požadován vždy bez rozdílu.[25]

**7.1.1.3 Platební karta** Platby platebními kartami jsou oblíbenou platební metodou v online světě. Mezi přednosti patří snadné použití, rychlé zpracování platby s potvrzením o přijetí, vysoká bezpečnost a bezpečnostní standard 3D Secure, které funguje na principu zaslání jednorázového ověřovacího kódu na mobilní telefon majitele karty.

Platbu platební kartou podporují všechny kartové asociace. Mezi nejznámější patří VISA a MasterCard.

**7.1.1.4 Bankovní převod** Bankovní převod je nejstarší způsob transakce financí na internetu. Platbu lze zadat kdykoliv prostřednictvím internetového bankovníctví, ale je zpracována pouze v úředních hodinách bankovní instituce. Jedná se o jednu z nejbezpečnějších platebních metod.

**7.1.1.5 Online bankovní tlačítko** Bankovní tlačítko je oblíbená metoda platby. Princip spočívá v tom, že se předvyplní platební příkaz v elektronickém bankovníctví. Transakce probíhá vždy v reálném čase v rámci jedné banky. Pro každé bankovní tlačítko musí mít obchodník založená účet u dané instituce.

Mezi výhody patří platba na jedno kliknutí, předvyplněný příkaz k úhradě eliminující chybovost při jeho zadávání a fakt, že platba je zadána v rámci vlastního internetového bankovníctví.

**7.1.1.6 Mobilní platba** Jedná se o platební metodu, při které je možné realizovat platbu z mobilního telefonu. Zaplacená částka je strhávána z tarifu zákazníka nebo předplacené karty. Mobilní platbu je možné zadat kdykoliv a odkudkoliv v reálném čase s garancí potvrzení provedení platby. Jsou využívány hlavně k platbám menších částek.

Příkladem jsou platby mPlatba, či Premium SMS.

**7.1.1.7 Elektronická peněženka** Elektronická peněženka patří mezi novější metody on-line plateb. Princip je následující. Uživatel si nabije kredit elektronické peněženky a z něj jsou následně placeny jednotlivé částky. Jedná se o alternativu ke klasickému bankovnímu účtu. Umožňuje svým registrovaným zákazníkům provádět transakce bezpečně a nevyžaduje oproti bankovnímu účtu ověřování osobních údajů klientů.

Průkopníkem této metody je služba PayPal. Mezi výhody patří jednoduchost, malá vysledovatelnost a okamžitá úhrada dané částky.

**7.1.1.8 Digitální mobilní peněženka** Základními předpoklady pro tento způsob placení je vlastnictví mobilu nebo tabletu, internetového připojení v mobilu (mobilní data nebo Wi-Fi) a platební kartu s aktivovanou možností platby na internetu. Princip digitální mobilní peněženky spočívá v tom, že si zákazník zadá informace o svých kreditních kartách do mobilní aplikace a přes ní může následně platit online i v kamenných prodejnách u vybraných obchodníků. Při placení již nepotřebuje mít fyzicky platební kartu u sebe.

Mezi výhody digitální peněženky patří pozbytí nutnosti opisování údajů z karty a kódů z SMS, platba více kartami, rychlá platba jedním klikem nebo vyfocením QR kódu. Metoda je podporována karetními asociacemi MasterCard a VISA.

**7.1.1.9 Kuponová platba** U kuponové platby probíhá úhrada hotovostí nebo předplatným skrze síť platebních terminálů. Mezi výhody patří vysoká bezpečnost, možná absence připojení k internetu, potvrzení a zadání platby v reálném čase.

Metodu využívají mladé systémy, jakým je superCASH, který umožňuje platit kupony s čárovými kódy prostřednictvím terminálů společnosti Sazka nebo České pošty. Dalším zástupcem je paysafecard, jenž umožňuje platbu přes předplacenou kartu, kterou je možné získat na benzínových pumpách a pobočkách sázkových kanceláří. Při využití této služby je platba potvrzena zadáním číselného kódu.

**7.1.1.10 Platba virtuální měnou** Virtuální měny se těší stále větší oblíbenosti mezi zákazníky. Nejznámější a neoblíbenější virtuální měnou je bitcoin. Platba virtuální měnou probíhá převodem z jedné peněženky virtuální měny do druhé.

Mezi výhody patří okamžité připsání platby a snadná směnitelnost za prakticky všechny měny na světě. Největším benefitem je naprostá anonymita. Nevýhodou je, že neexistuje žádný proces vrácení platby například při podvodech, jak bývá zvykem u zavedených platebních metod. Mezi nevýhody virtuálních měn je extrémní volatilita, tedy velké kolísání měnového kurzu. V Česku platbu bitcoinem umožňuje například jeden z největších e-shopů Alza.cz.[22]

## **7.1.2 Obchodní model portálu Gloffer**

Webový portál Gloffer poskytuje většinu funkcionalit zcela zdarma. V první fázi portálu Gloffer bude omezována služba cenového srovnávače, přístup k poptávkám poptávkového systému a

funkce automatického prodejce. Do budoucna systém počítá s možností zpoplatnění reklamních prostor portálu.

Budoucí zákazník má na výběr z paušálové platby nebo platby za zrealizovaný obchod. Vzhledem k tomu, že se portál nachází ve spouštěcí fázi, nabízí prvním 2000 firmám možnost doživotně využívat služby cenového srovnávače a automatického prodejce zcela zdarma. Pro ověření rentability řešení je firemním klientům umožněno zrealizovat první tři obchodní případy zcela zdarma.

**7.1.2.1 Paušálová platba** U paušálových plateb je hlavním ukazatelem časové období zakoupeného paušálu. Pokud má uživatel zakoupenou danou službu v daném období, je mu povolen přístup k této službě. Dle aktuálního ceníku je možné zvolit měsíční nebo roční paušál.

**7.1.2.2 Platba za zrealizovaný obchod** Druhou variantou je využití zpoplatnění jednotlivých transakcí. Pro tuto variantu si musí uživatel nejprve dobít kredit do systémové peněženky. Z této peněženky jsou následně strhávány poplatky za poskytované nebo zrealizované služby. Firma si vybírá z varianty fixní platby za zrealizovaný obchod nebo procento z prodeje.

## 7.2 Návrh a implementace

V této části práce jsem se věnoval návrhu a implementaci kontroly oprávnění uživatelů, procesu objednání služeb, metodám zaplacení objednávky a omezení systému na základě těchto objednávek.

Součástí aplikace je také správa objednávek a s tím související akce. Dle pokynů k vypracování jsem také generoval faktury k objednávkám ve formátu PDF.

Pro aktivaci zakoupených služeb při zvolené platební metodě platbou na účet je zapotřebí nejprve zajistit ověření, zdali byla objednávka uhrazena. Z tohoto důvodu jsem implementoval komponentu, která zajišťovala komunikaci s bankou Fio pro účely stahování transakcí. Jelikož druhou variantou úhrady faktury je platba platební kartou, vytvořil jsem komponentu, která zajišťovala komunikaci s API platební brány. Z možných platebních bran na trhu byla zvolena brána GoPay.

Česká legislativa nařizuje při platbě kartou online vystavovat EET účtenky. Systém bude pracovat pouze s jednou sazbou DPH. Z toho důvodu nebylo nutné neimplementovat vlastní napojení na Finanční správu České republiky, ale připadalo v úvahu použití existujícího řešení vystavování EET účtenky prostřednictvím platební brány GoPay. Tuto variantu jsem také zvolil.

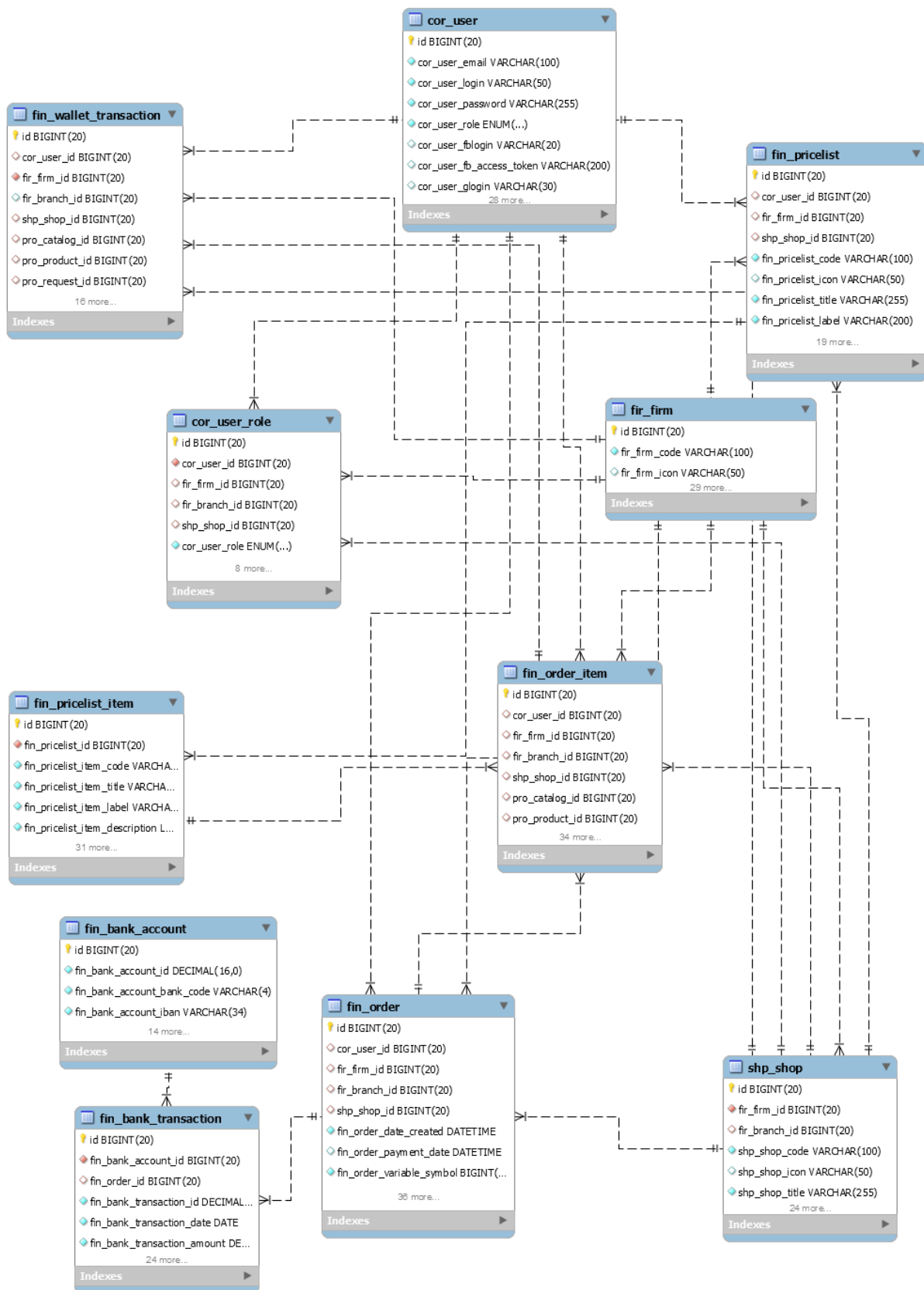
Po zaplacení služeb následuje proces kontrola a omezování zakoupených služeb.

### 7.2.1 Databázový návrh

Návrh databázové struktury se skládá z entit pro uložení ceníku, objednávky, zakoupené služby a bankovních transakcí (Obrázek 27). Dále obsahuje entity reprezentující role uživatelů portálu.

Seznam použitých tabulek:

- fin\_pricelist – Ceník
- fin\_pricelist\_item – Položka ceníku
- fin\_order – Objednávka
- fin\_order\_item – Položka konkrétní objednávky
- fin\_wallet\_transaction – Transakce v peněžence
- fin\_bank\_account – Bankovní účet
- fin\_bank\_transaction – Transakce bankovních účtů
- cor\_user – Uživatel systému
- fir\_firm – Firma v systému
- shp\_shop – Obchod v systému
- cor\_user\_role – Uživatelská práva uživatelů k firmě a obchodu



Obrázek 27: ER diagram ceníků, objednávek a transakcí

### 7.2.2 Oprávnění uživatelů

Kontrola oprávnění a omezení přístupu pro firemní klientelu systému je nezbytným krokem poskytování služeb. V systému existují tři základní role. V rámci kontroly oprávnění je ověřováno, zdali daný uživatel má přístup k firemnímu nebo e-shopovému profilu a objektům pod správou tohoto profilu. Spravovat oprávnění je možné v sekci Správa oprávnění.

**7.2.2.1 Role uživatelů** V systému jsou definovány tři role uživatelů, které ovlivňují přístup k firmě nebo e-shopu:

- user – Uživatel s právy vlastníka může ve většina případů pouze nahlížet na danou firmu nebo e-shop a všechny objekty s ním spojené.
- manager – Tato role dovoluje uživateli provádět veškeré operace s firmou nebo e-shopem. Umožňuje také editovat, přidávat a mazat objekty, které daná firma nebo e-shop vlastní nebo spravuje. Uživatel v této roli může přidávat a odstraňovat oprávnění ostatních uživatelů. Nemůže ale měnit vlastníka firmy.
- owner – Role owner určuje uživatele, který danou firmu nebo e-shop zaregistroval. Práva má stejná jako role manager.

**7.2.2.2 Správa oprávnění** Tato část systému slouží k přehledu uživatelů s přístupovými právy do dané firmy nebo e-shopu (Obrázek 28). Jednotlivá oprávnění je možné změnit nebo odebrat. Tyto operace jsou dostupné pro uživatele v roli owner nebo manager.

SPRÁVA OPRAVNĚNÍ PRO FIRMU: (TESTOVACÍ FIRMA)					
+ Přidat oprávnění					
Uživatel	Email	Role	Vytvořeno	Změněno	Akce
Novák Martin	novak@gloffer.com	Vlastník	8.1.2018 14:43:48	8.1.2018 15:43:48	
Mičanin Jan	micanin@gloffer.com	Manažér	9.1.2018 10:54:07	9.1.2018 11:54:07	Vaše aktuální oprávnění
Poklík Václav	poklik@seznam.cz	Manažér	9.1.2018 22:35:25	31.1.2018 14:21:49	
Lasuša František	lasuga@glof.com	Manažér	13.2.2018 15:49:25	13.2.2018 16:49:25	

Obrázek 28: Správa oprávnění přístupu k firmě

**7.2.2.3 Kontrola oprávnění** Kontrolu oprávnění ke všem objektům v systému má na starost třída s názvem *ObjectAuthorizator*. Oprávnění je dvojího typu. Přístup k objektu bez možnosti akcí zajišťuje metoda *hasUserRole*, která ověřuje existenci jakékoliv role k objektu. Pro



kontrolu přístupu k operacím, které ovlivňují dané objekty slouží metoda *hasAdminRole*. Funkce ověřuje, zdali má uživatel roli vlastník nebo manažér.

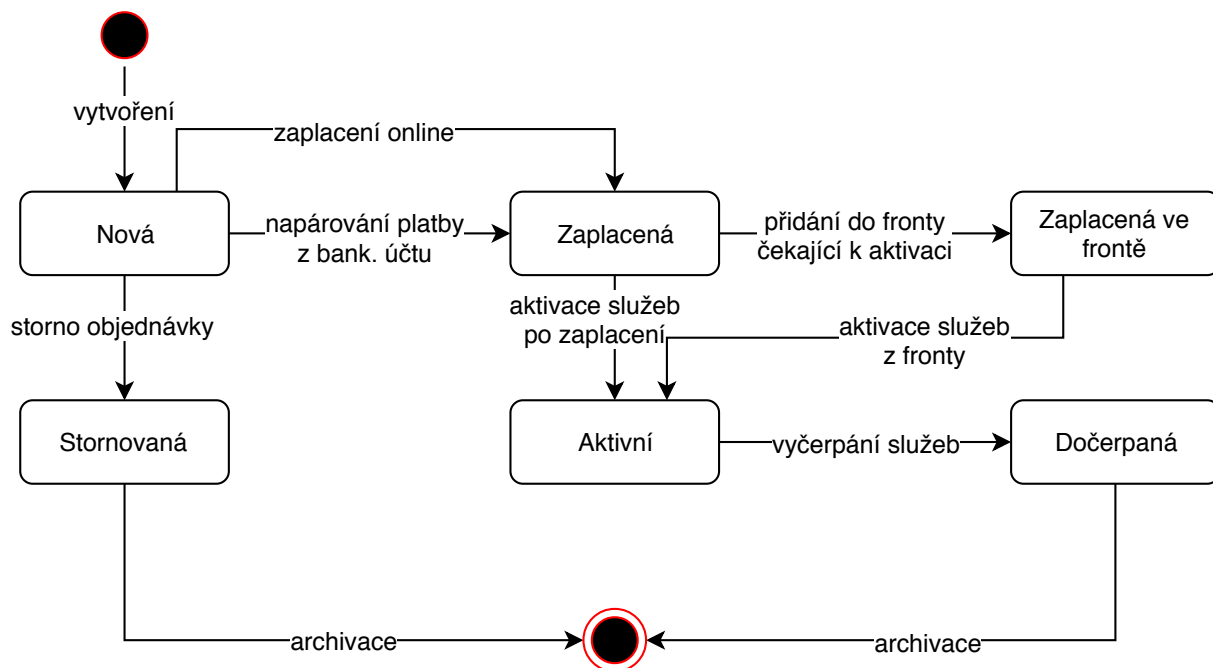
V situaci, kdy uživatel nemá dostatečná oprávnění k dané firmě nebo e-shopu či konkrétnímu objektu, je vyvolána výjimka *ForbiddenRequestException*, která způsobí zablokování přístupu.

### 7.2.3 Objednávání služeb

Proces objednávání služeb začíná objednáním služeb vytvořením ceníku. Následuje vyplnění objednávkového formuláře, ve kterém je možné vybrat balíček služeb z ceníku. Dalším krokem je samotné vytvoření objednávky a kontrola zaplacených služeb a jejich limitů.

Při návrhu ceníku (Obrázek 29) jsem se inspiroval existujícími řešeními na trhu. Po jeho definování následovala implementace komponenty pro vizualizaci ceníku. Součástí této komponenty je také formulář sloužící pro vytvoření objednávky uživatelem. Vizualizace ceníku je dostupná pro všechny přihlášené uživatele. Objednávání služeb je dostupné pouze v roli Firma a E-shop.





Obrázek 30: Stavový diagram objednávky

#### 7.2.4 Vygenerování faktury do PDF

Pro zaplacení objednaných služeb je zapotřebí nejprve vygenerovat fakturu. Ta je v systému vytvořena ve formátu PDF (Obrázek 31). Pro její vytváření jsem použil Nette knihovnu *pdfresponse*<sup>9</sup> od autora *jkuchar*. Komponentu jsem instaloval skrze Composer.

<sup>9</sup>PdfResponse (<https://github.com/jkuchar/PdfResponse>) ze dne 12.4.2018



FAKTURA #15900093

#### Dodavatel

Serious Investment s.r.o.,

Stanislavského 355, Ostrava - Svinov, 721 00

Česká republika

IČ: 29456568, DIČ: CZ29456568

Zapsán do OR: KS v Ostravě, C.54906

#### Odběratel

Testovací firma

17. listopadu 203/2, 708 00 Ostrava

Česká republika

IČ: 46545456465546 DIČ: 154546456546546

#### Banka

#### Fio banka, a.s.

#### Bankovní účet

2900951660/2010 (CZK)

#### IBAN

CZ8520100000002900951660

#### BIC/SWIFT

FIOBCZPPXXX

#### Variabilní symbol

15900093

#### Forma úhrady

Kreditní karta

#### Datum vystavení

9. 4. 2018

#### Datum splatnosti

23. 4. 2018

#### Datum uskutečnění zdan. plnění

13. 4. 2018



ZAPLACENO

Položka	Ks	Cena/ks	Cena	Sazba DPH	Celkem s DPH
Balíček služeb - Paušál roční	1	1 800 Kč	1 800 Kč	21%	2 178 Kč
<b>Celkem</b>			1 800 Kč	21%	<b>2 178 Kč</b>

#### Souhrn informací o platbě

Celkem k úhradě

0 Kč

info@gloffer.com, +420 603 90 10 90

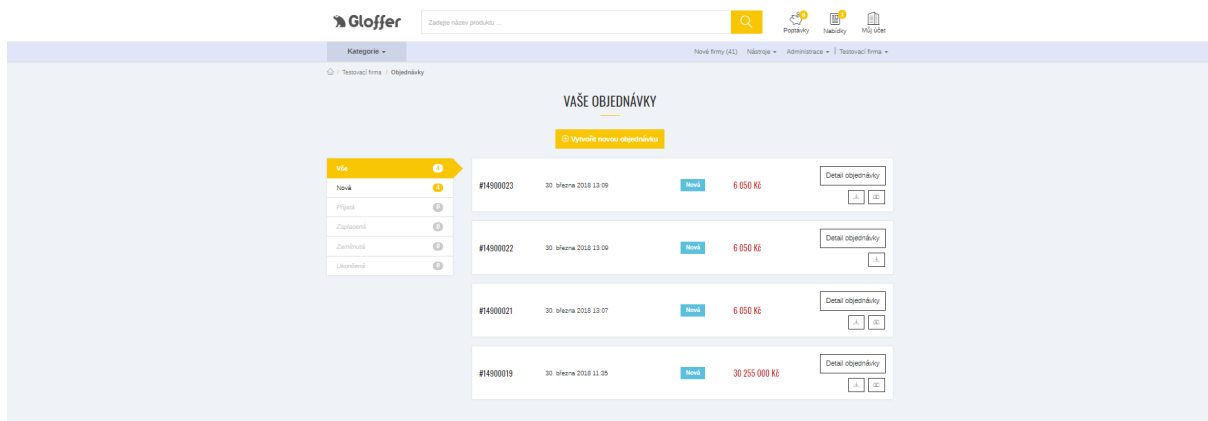
Tato objednávka byla vystavena automaticky, v případě dotazů či nejasností nás neváhejte kontaktovat.

Obrázek 31: Faktura ve formátu PDF

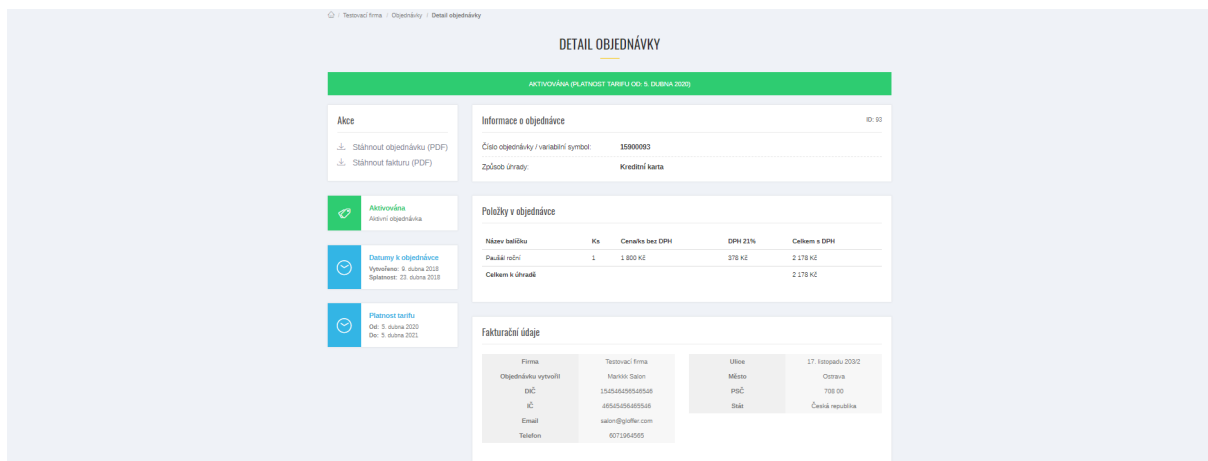
### 7.2.5 Správa objednávek

Správa objednávek dostupná uživatelům v roli Firma a E-shop. Základem jsou stránky s přehledem objednávek (Obrázek 32) a jejich detaily (Obrázek 33). Správa objednávek slouží jako prostředek pro manipulaci a kontrolu objednávek.

Z vizuálního pohledu se v hlavní části seznamu objednávek nachází jejich základní informace. V další části jsou umístěny odkazy pro generování PDF faktury a zaplacení objednávky online. Na stránce s detailem se nachází podrobné informace o objednávce a stejná akční tlačítka jako v přehledu objednávek.



Obrázek 32: Přehled objednávek služeb



Obrázek 33: Detail objednávky služeb

### 7.2.6 Získávání informací o transakcích z bankovního účtu

Získávání informací o transakcích z bankovního účtu slouží pro změnu stavu objednávek a aktivaci zakoupených služeb. Před samotnou komunikací s API je nejprve nutné získat token umožňující přístup k danému bankovnímu účtu. O token lze zažádat v internetovém bankov-

nictví Fio. Samozřejmostí je také možnost nastavení práv a operací, které je možné pro daný token provádět. V rámci mé práce byl dostatečný token s oprávněními, která umožňovala pouze získávání informací o zrealizovaných transakcích. Přes tento typ tokenu není možné zadávat hromadné platby účtu. Tuto funkci poskytuje pouze token s neomezeným přístupem.

Pro spojení s Fio jsem implementoval sadu funkcí komunikující s API banky. Jedná se především o funkce zabezpečující stahování nových transakcí a jejich ukládání do databáze. Další funkcionalitou této sady je spárování platby na základě variabilního symbolu a ceny. Pokud se tyto dva parametry přesně neshodují, nedojde k automatickému spárování s objednávkou. Párování platby probíhá automaticky v daných časových intervalech.

Metoda pro získávání informací o transakcích se jmenuje *putTransactionsFromBankAccountToDB* (Výpis 8) a nachází se ve třídě *PaymentManager*. Pro získání odpovědi z API jsem využil nativní PHP funkci *file\_get\_contents*. Alternativou je využití příkazu *cURL*.

---

```
if ($type == "all") {
    $fioAPIDateFrom = "2017-12-01";
    $fioAPIDateTo = date('Y-m-d');
    $fioAPIUrl = "https://www.fio.cz/ib_api/rest/periods/". $this->bankAccount
        ->token. "/". $fioAPIDateFrom ."/". $fioAPIDateTo ."/transactions.json";
}
elseif ($type == "new"){
    $fioAPIUrl = "https://www.fio.cz/ib_api/rest/last/". $this->bankAccount->
        token. "/transactions.json";
}

$fileContent = file_get_contents($fioAPIUrl);
if ($fileContent == false) {
    $this->logger->log(new \Exception("Fio API neodpovd."), ILogger::EXCEPTION)
        ;
    return false;
}

$this->fioAPIResponse = json_decode($fileContent,true);

if ($this->fioAPIResponse['accountStatement']['info']['iban'] != $this->
    bankAccount->iban ) {
    $this->logger->log(new \Exception("sla t se neshoduj."), ILogger::
        EXCEPTION);
    return false;
}
```

```

foreach ($this->fioAPIResponse['accountStatement']['transactionList']['
transaction'] as $item) {
    $this->transactionManager->transactional(function () use ($item){
        /** @var BankTransaction $transaction */
        $transaction = new BankTransaction();
        $transaction->bankAccount = $this->bankAccount;
        $transaction->type = $item[self::TRANSACTION_TYPE]['value'];
        $transaction->moveId = $item[self::TRANSACTION_MOVE_ID]['value'];

        if (isset($item[self::TRANSACTION_SS]['value'])) {
            $transaction->specificSymbol = $item[self::TRANSACTION_SS]['value'];
        }

        if (isset($item[self::TRANSACTION_COMMENT]['value'])) {
            $transaction->comment = $item[self::TRANSACTION_COMMENT]['value'];
        }

        /** @var Order $order */
        $order = $this->orderRepository->tryFind(["fin_order_variable_symbol"
=> $transaction->variableSymbol, "fin_order_status" => [Order::
STATUS_NEW, Order::STATUS_ACCEPTED]]);

        if ($order !== false && $order->totalPrice == $transaction->amount &&
$order->currency == $transaction->currency) {

            $transaction->order = $order;
            $order->status = Order::STATUS_PAYED;
            $order->paymentDate = new \DateTime();
            $this->orderRepository->persist($order);
        }
    });
}
return true;

```

---

Výpis 8: Ukázka části těla metody pro stahování transakcí z Fio účtu

### 7.2.7 Platba kartou pomocí GoPay platební brány

Při online platbě přes platební bránu existují dvě možnosti. První je integrace okna platební brány přímo do stránky obchodníka. Druhou možností je přesměrování uživatele na stránku platební brány. Při mé implementaci jsem využil druhou možnost z důvodu vyššího zabezpečení. S platební bránou GoPay se komunikuje prostřednictvím jejího API.

Před zahájením samotné komunikace s API je zapotřebí nejprve požádat o přístup k němu. Žádost se zasílá přes webové stránky GoPay. Po schválení žádosti následuje poskytnutí přístupových údajů pro testovací verzi brány.

Pro komunikaci s GoPay API jsem využil SDK poskytované společností GoPay.<sup>10</sup> Tato knihovna byla také, podobně jako ostatní použité knihovny, nainstalována přes službu Composer.

Prvním krokem komunikace s API je konfigurace připojení. Poté následuje získání tokenu, který slouží k ověření komunikace. Délka platnosti tokenu je 30 minut. Token je nutné uvádět vždy v hlavičce požadavku. Dalším krokem platby kartou je založení samotné platby. Při zakládání dochází ke specifikaci platební metody, zadání informací o plátcí, měna transakce, cena uvedená v haléřích, číslo objednávky, popis objednávky, položky objednávky a návratová URL, na kterou je uživatel přesměrován při návratu ze serveru platební brány. Na tuto URL je také zaslána po provedení platby notifikace o jejím stavu. Tato notifikace byla následně zpracována a případně změněn stav objednávky a povolení zakoupené služby. Součástí upozornění jsou také údaje EET. V odpovědi na požadavek o založení platby jsou nejdůležitější parametry *ID* (identifikátor transakce) a *gw\_url* (URL, na kterou proběhne přesměrování zákazníka pro zadání platby). Následuje samotné přesměrování na platební bránu, provedení platby a navrácení se na stránku obchodníka.

Pro případ změny stavu platby mimo samotný proces provedení platební operace existuje metoda, která zpracovává asynchronní notifikace zasílané platební branou.

O proces založení platby se stará metoda *createNewPayment*. Stav platby je kromě notifikace při přesměrování z platební brány možné také získat pomocí metody *getPaymentStatus*. Obě základní funkce se nacházejí ve třídě *GoPayManager*. [24]

Podrobné informace potřebné pro integraci jsou dostupné na stránkách platební brány.<sup>11</sup>

### 7.2.8 Omezování zakoupených služeb

Omezování zakoupených služeb je posledním krokem procesu objednávky služeb. V návaznosti na obchodní model portálu Gloffer existují dva typy tarifů (paušální a kreditový 7.1.2), na jejichž základě je kontrolován přístup k placené službě.

V případě souběhu časově omezeného a kreditního tarifu je prioritizován tarif časově omezený. Proces kontroly přístupu se liší podle typu zakoupeného tarifu.

<sup>10</sup>GoPays's PHP SDK (<https://github.com/gopaycommunity/gopay-php-api>) ze dne 16.4.2018

<sup>11</sup>Integrace GoPay (<https://gopay.com/cs/integrace.html>) ze dne 16.4.2018



Pro účel kontroly přístupu slouží třída *ServiceAuthorizator*. Ta má také na starosti zaúčtování transakcí a nabití kreditu systémové peněženky u kreditních tarifů.

**7.2.8.1 Časově omezené tarify** Každý balíček služeb zakoupený na určité období má jasné definovanou platnost. V případě, že dojde k objednávce a zaplacení balíčku v době, kdy je aktivní jiný paušální balíček, dochází k posunutí data začátku na konec aktivního balíčku.

---

```
SELECT 'fin_order_item'.*
FROM 'fin_order_item'
WHERE duration_begin <= NOW() AND duration_end >= NOW() AND 'fir_firm_id' = 210
AND
'fin_service_demander' = 'yes' AND active IN ('new', 'enabled')
ORDER BY duration_begin ASC
LIMIT 1
```

---

Výpis 9: Dotaz časově omezené paušalovými tarify

**7.2.8.2 Kreditové tarify** U kreditových služeb není nastavené datum konce platnosti. Tarif je ukončen v době, kdy dojde k zápornému zůstatku v peněžence. Pokud je zakoupen kreditový tarif ve chvíli, kdy je aktivní jiný kreditový tarif, je nově zakoupený zařazen do fronty ke zpracování. Po vyčerpání aktuálně běžícího kreditu je automaticky aktivován další v pořadí.

---

```
SELECT 'fin_order_item'.*
FROM 'fin_order_item'
WHERE duration_begin IS NOT NULL AND duration_end IS NULL AND 'fir_firm_id' =
210 AND
'fin_service_demander' = 'yes' AND active IN ('new', 'enabled')
ORDER BY duration_begin ASC
LIMIT 1
```

---

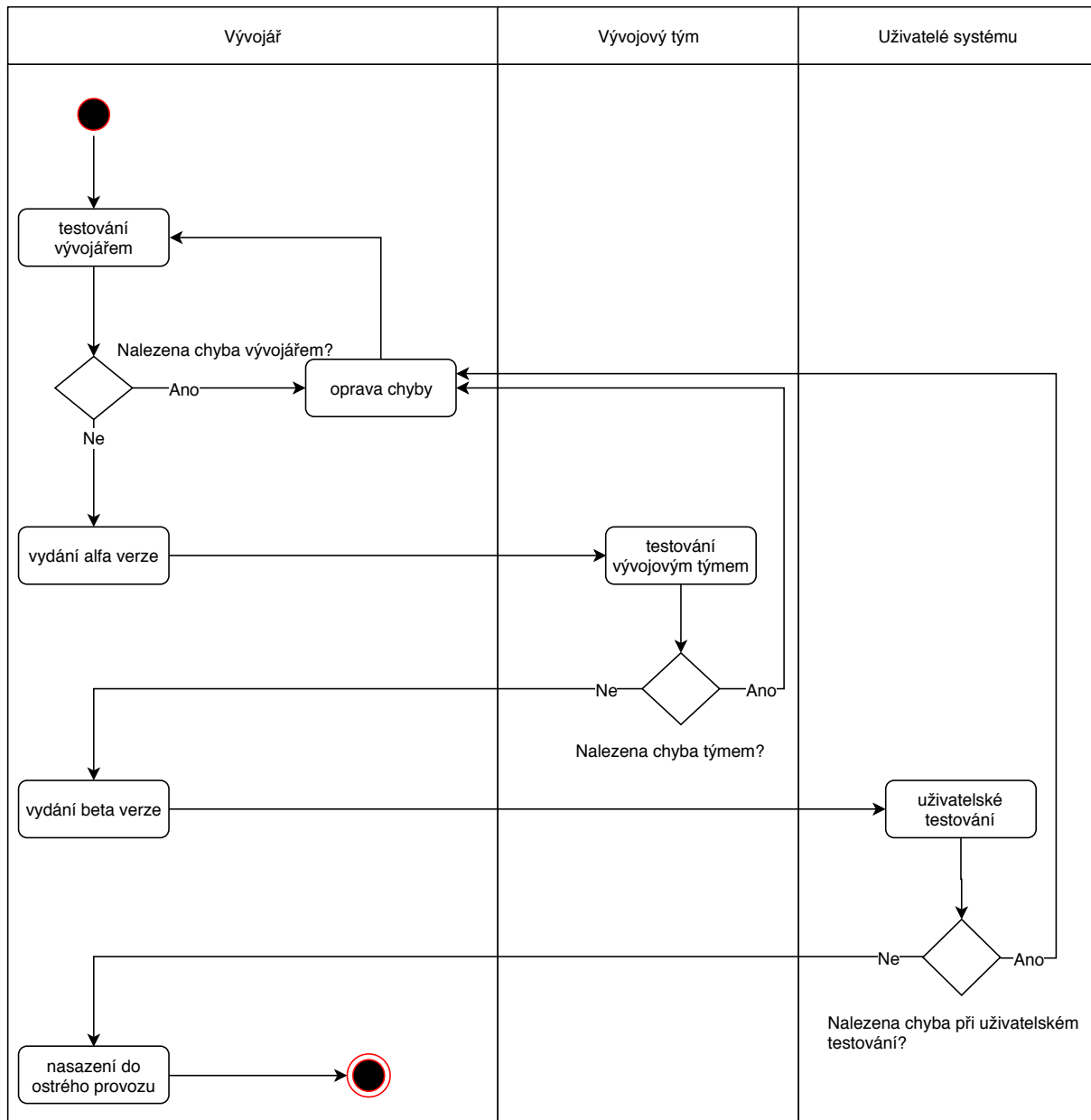
Výpis 10: Dotaz na omezené kreditovými tarify

**7.2.8.3 Zablokování služby** Po vypršení platnosti časového tarifu (Výpis 9) nebo překročení nabitého kreditu (Výpis 10) dojde k blokaci přístupu ke službě pro danou firmu nebo e-shop. V rámci služby Poptávkový systém je umožněno dokončit rozpracované obchody. Nové obchodní případy již dále vytvářet není možné.

## 8 Testování a nasazení

Po dokončení implementace jednotlivých modulů proběhlo několik fází testování a nasazení.<sup>34</sup>

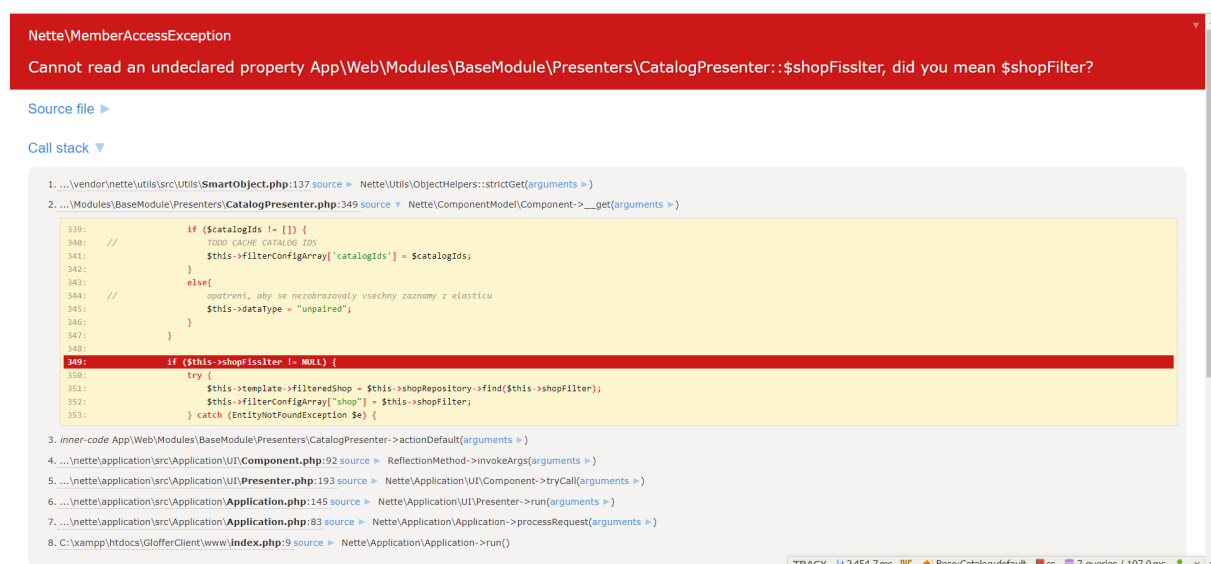
V první fázi jsem prováděl vývojářské testování veškerých možných stavů a situací, které mohou v programu nastat. Po mém testování bylo provedeno uživatelské testování ostatními členy vývojového týmu portálu Gloffer.



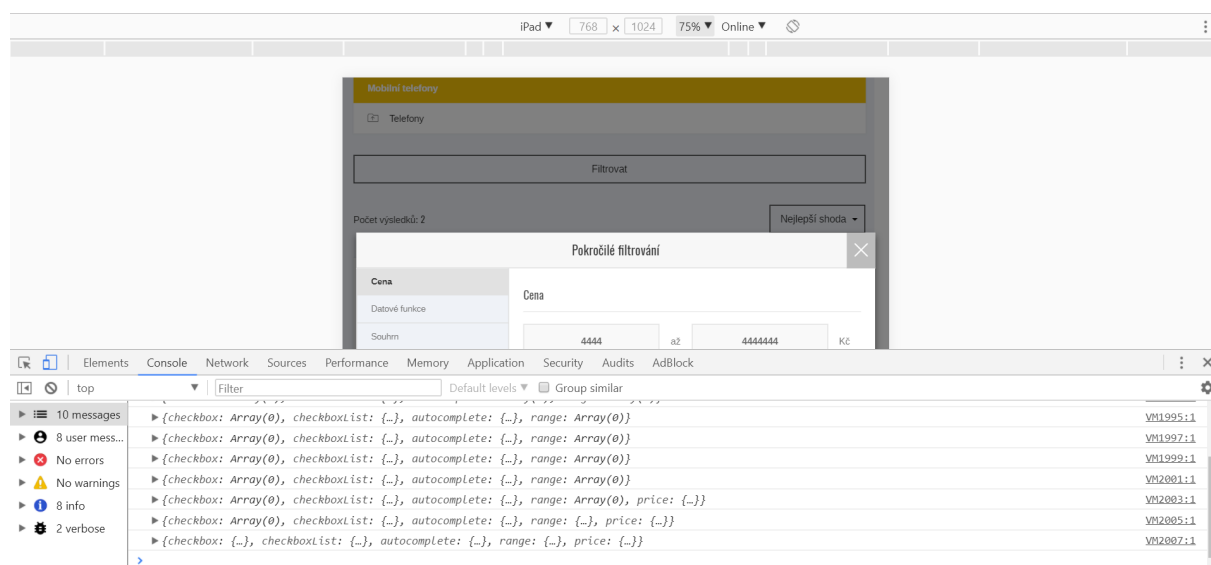
Obrázek 34: Diagram testování a nasazení aplikace

K debugování zdrojového kódu v Nette jsem využil nativní ladící nástroj frameworku Tracy (Obrázek 35). Ajaxová komunikace byla odladěna pomocí developerské konzole internetového

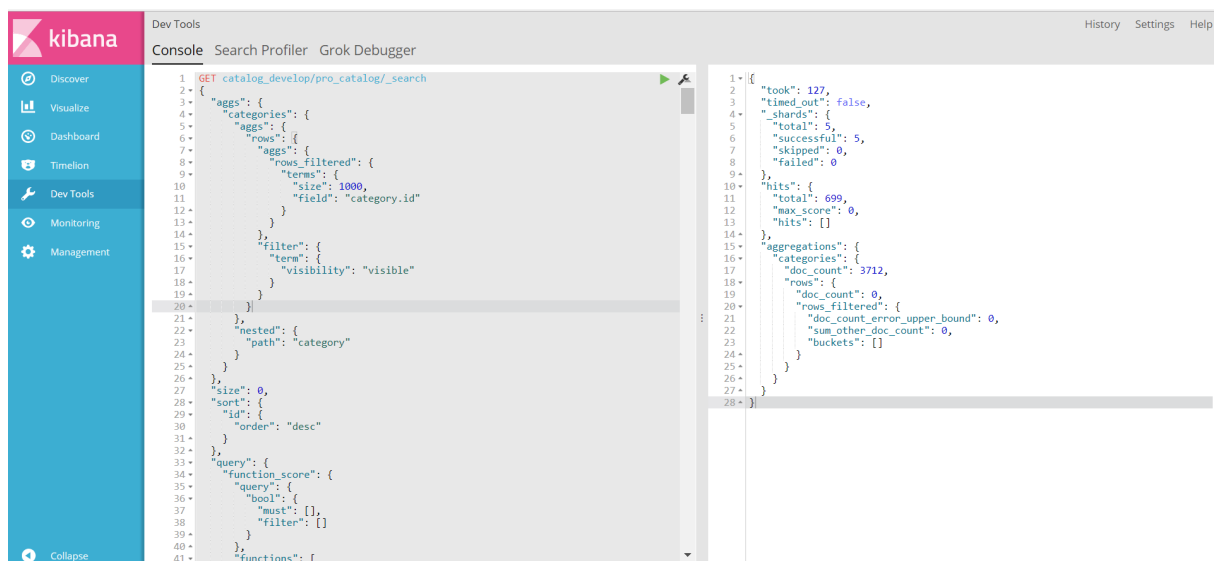
prohlížeče Google Chrome (Obrázek 36) a Mozilla Firefox. V tomto prostředí probíhalo také ladění programové části aplikace, která běží na straně klienta a je napsaná v jazyce Javascript a knihovně JQuery. K ladění a testování dotazů vyhledávání byla použita developerská konzole monitorovacího systému Kibana (Obrázek 37).



Obrázek 35: Ladící nástroj Tracy



Obrázek 36: Developerské prostředí internetového prohlížeče Google Chrome



Obrázek 37: Developerská konzole monitorovacího systému Kibana

Veškerá funkcionalita byla otestována napříč běžně používanými internetovými prohlížeči. Samozřejmostí bylo také testování na mobilním zařízení.

Implementace jednotlivých modulů probíhala pomocí verzovacího systému na větvích oddělených od hlavní vývojové větve projektu Gloffer. Po dokončení fáze testování členy vývojového týmu proběhlo spojení těchto větví do hlavní větve. Následovalo další uživatelské testování beta verze na vzorku 20 osob.

Poslední fází nastala po dokončení celkového testování beta verze. V této fázi proběhlo nasazení do ostrého provozu.

## 9 Možnosti dalšího rozvoje řešení

I přes fakt, že se podařilo splnit všechny body zadání této práce, je zde prostor pro budoucí vylepšení. V době vzniku práce mě osobně napadla následující vylepšení.

V modulu parametrického vyhledávání by bylo uživatelsky přívětivější variantou zobrazování výsledků s využitím ajaxového požadavku a dynamického překreslení stránky. U jednotlivých vlastností parametrů je možné přidat informaci o tom, kolik produktů bude z výsledku vyhledávání přidáno nebo odebráno vybráním dané hodnoty vlastnosti. Pro zlepšení uživatelského používání je zapotřebí také provést seřazení a prioritizaci kategorií, produktů, vlastností a hodnot vlastností. Automatizace procesu řazení lze realizovat na základě logování navštívených stránek a vytvořených filtrů reálnými uživateli systému. Pro shromažďování údajů je možné využít analytické nástroje, pro příklad Google Analytics, nebo vlastní interní logování systému. Druhým způsobem řazení je využití lidské síly, který ale při malém počtu lidí v týmu není možný z kapacitních důvodů realizovat nad všemi kategoriemi katalogu. Pro zrychlení procesu generování datových podkladů je vhodné nadále pracovat na optimalizaci dotazů a databázových indexů.

V poptávkovém systému je do budoucna prostor pro rozvoj v realizaci dalších typů poptávek, které byly zmíněny v podkapitole 5.2.3.

V rámci tiketovacího systému je možné v budoucnu vytvořit přehledné dashboardy se statistickými daty. U odesílání emailu je zapotřebí vytvořit vlastní SMTP server, aby celý proces nebyl závislý na hostingovém poskytovateli.

V modulu online plateb a objednávek je velký potenciál v rozšiřování současné nabídky platebních metod. Další inovací může být komponenta pro automatické opakování objednávek. Účelem této funkcionality bude vytvoření objednávky z objednávky stávající. Objednávka bude vytvářena před ukončením platnosti stávajících služeb, a tím pádem nebude docházet k výpadkům služeb. Automatické opakování objednávek přinese komfortnější prostředí jak pro provozovatele, tak pro uživatele systému.

Součástí budoucího rozvoje bude také notifikační systém, který bude sloužit k upozornění uživatelů na různé události. Může se jednat například o nové poptávky, nabídky, hodnocení poptávek, komunikace v poptávkovém systému, zprávy z tiketovacího systému nebo upozornění na končící objednané služby.

## 10 Závěr

Cílem práce bylo zanalyzovat problematiku parametrického vyhledávání, poptávek, správy uživatelských požadavků a možností online plateb na internetu. Na základě analýzy následoval návrh a implementace vlastních řešení.

Při návrhu vlastního řešení a implementaci jsem využil zkušenosti a informace nabyté analýzou daných problematik. Další zkušenosti jsem získal během studia v předmětech Algoritmy vykonávání dotazů, Analýza rozsáhlých dat a Internetové technologie. S některými použitými technologiemi jsem se už setkal při tvorbě bakalářské práce, jiné bylo zapotřebí nově nastudovat.

Vytvořené parametrické vyhledávání, dle mého názoru, je na podobné kvalitativní úrovni jako již existující řešení na trhu. Vzhledem k faktu, že veškerá existující řešení jsou komerčního charakteru a jejich implementace nejsou veřejně dostupné, nemohlo být provedeno přímé srovnání použitých technologií a porovnání výkonu. Navržené a vytvořené řešení správy uživatelských požadavků a komunikace se zákazníky v porovnání s existujícími ticketovacími systémy nedosahuje kvality zavedených řešení na trhu, avšak plně odpovídá aktuálním potřebám portálu Gloffer. Nebude zvyšovat náklady způsobené využitím aplikací třetích stran, sjednotí firemní komunikaci se zákazníky a umožní získávání cenné zpětné vazby od uživatelů. U procesu objednávání a hlídání zakoupených služeb bylo nutné vytvořit vlastní řešení, jelikož se jedná o velmi specifickou oblast, kterou není možné nahradit použitím existujícím řešením na trhu.

Většina vlastního řešení se podařila úspěšně nasadit do reálného provozu. Výjimkou je pouze převedení online plateb z testovacího do provozního režimu, které nebylo možné provést z důvodu nesplnění všech podmínek migrace v době vzniku této práce kladené na obchodní místo ze strany GoPay. Komunikace s platební bránou byla provedena jen v testovacím režimu. Po dokončení práce také došlo ke zrušení povinnosti evidování EET u online plateb platební kartou.

Během vytváření práce byly také nalezeny nové potřeby a požadavky k řešení, které nebyly definovány v zadání této práce. I ty se podařilo úspěšně zakomponovat do finální verze aplikace.

Největší radostí a zdrojem informací při tvorbě této práce bylo působení v perspektivním a profesionálním vývojovém týmu poptávkového portálu Gloffer. Nejtěžším úkolem bylo spojení relativně rozličných bodů zadání této práce do jednoho uceleného díla. Tento úkol se, z mého pohledu, nakonec podařilo splnit.

## Literatura

- [1] Nette [online], Nette Foundation [cit. 2018-02-12]. Dostupné z: <https://nette.org/>
- [2] Composer [online], [cit. 2018-02-12]. Dostupné z: <https://getcomposer.org/>
- [3] Git [online], [cit. 2018-02-12]. Dostupné z: <https://git-scm.com/>
- [4] MySQL [online], [cit. 2018-02-13]. Dostupné z: <https://www.mysql.com/>
- [5] Open Source Search & Analytics Elasticsearch [online], [cit. 2018-02-13]. Dostupné z: <https://www.elastic.co/>
- [6] GORMLEY, Clinton a Zachary TONG, 2015. Elasticsearch: the definitive guide. Sebastopol, CA: O'Reilly. ISBN 14-493-5854-3.
- [7] Redis [online], [cit. 2018-02-13]. Dostupné z: <https://redis.io/>
- [8] Fio banka - česká banka pro váš účet nebo investice | Fio banka, API Bankovníctví | Fio banka[online]. [cit. 2018-02-20]. Dostupné z: <https://www.fio.cz/bankovni-sluzby/api-bankovnictvi>
- [9] GoPay REST API, Platební brána pro váš byznys | GoPay [online]. [cit. 2018-02-23]. Dostupné z: <https://doc.gopay.com/cs/>
- [10] ING. DURČÁK, Pavel, Čárové kódy EAN | NaPočítači.cz. In: NaPočítači.cz [online]. 25.7.2017 [cit. 2018-02-25]. Dostupné z: <https://www.napocitaci.cz/33/carove-kody-ean-uniqueidgOkE4NvrWuNY54vrLeM676VDceAa10MHAgrzFJ8D5c>
- [11] Alza.cz - největší obchod s počítači a elektronikou [online], [cit. 2018-03-01]. Dostupné z: <https://www.alza.cz/>
- [12] CZC.cz - rozumíme vám i elektronice [online], [cit. 2018-03-01]. Dostupné z: <https://www.czc.cz/>
- [13] Porovnání cen a srovnání produktů z internetových obchodů [online], [cit. 2018-03-01]. Dostupné z: <https://www.heureka.cz/>
- [14] Zboží.cz | Tisíce obchodů na jednom místě [online], [cit. 2018-03-01]. Dostupné z: <https://www.zbozi.cz/>
- [15] What is a Full-Text Search? - Definition from Techopedia, In: Techopedia - Where Information Technology and Business Meet [online]. [cit. 2018-01-10]. Dostupné z: <https://www.techopedia.com/definition/17113/full-text-search>
- [16] In: Algolia [online]. [cit. 2018-03-05]. Dostupné z: <https://www.algolia.com/doc/guides/getting-started/what-is-algolia/>

- [17] Memcached vs. Redis Comparison, DB-Engines [online]. [cit. 2018-03-05]. Dostupné z: [https://db-engines.com/en/ranking\\_trend/system/Memcached;Redis](https://db-engines.com/en/ranking_trend/system/Memcached;Redis)
- [18] Základní pojmy, In: Univerzitní informační systém MENDELU [online]. [cit. 2018-03-05]. Dostupné z: [https://is.mendelu.cz/eknihovna/opory/zobraz\\_cast.pl?cast=51387](https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=51387)
- [19] Customer Support Software & Ticketing System | Freshdesk [online], [cit. 2018-03-10]. Dostupné z: <https://freshdesk.com/>
- [20] Zendesk | Customer Service Software & Support Ticket System [online], [cit. 2018-03-10]. Dostupné z: <https://www.zendesk.com/>
- [21] Jira | Software pro sledování požadavků a projektů | Atlassian, Atlassian | Nástroje pro vývoj softwaru a spolupráci [online]. [cit. 2018-03-12]. Dostupné z: <https://cs.atlassian.com/software/jira>
- [22] Beyond | GoPay - Help, Platební brána pro váš byznys | GoPay [online]. [cit. 2018-03-15]. Dostupné z: <http://help.gopay.com>
- [23] Zásady komunikace se zákazníky - Shoptet.cz blog, Internetový obchod a pokladní systém zdarma! Pronájem u Shoptet.cz [online]. [cit. 2018-03-10]. Dostupné z: <https://blog.shoptet.cz/zasady-komunikace-se-zakazniky/>
- [24] Blog GoPay [online], [cit. 2018-03-12]. Dostupné z: <https://blog.gopay.com/>
- [25] PCI Security Standards [online], [cit. 2018-03-08]. Dostupné z: <http://www.pcistandard.cz/>
- [26] Digitální peněženka Masterpass by Mastercard, ČSOB [online]. [cit. 2018-03-13]. Dostupné z: <https://www.csob.cz/portal/lide/produkty/platebni-karty/doplujici-informace/digitalni-penezenka>



## A Příloha na CD/DVD

Přílohou na disku je elektronická verze textu diplomové práce a zdrojové kódy. Pro lepší orientaci uvádím strukturu složek na disku.

